SAIC-96/1179

New Region-Dependent Travel-Time Handling Facilities at the IDC; Functionality, Testing and Implementation Details

Submitted to:

Configuration Control Board
Center for Monitoring Research
1300 N. 17th Street, Suite 1450
Arlington, VA 22209

Prepared by:

Walter Nagy
Science Applications International Corporation
10260 Campus Point Drive
San Diego, CA 92121

July 30, 1996

OVERVIEW

This memo outlines functionality, testing and implementation issues related to the new travel-time handling facilities that have been developed by SAIC, San Diego, as part of the larger IDC Knowledge Acquisition (KA) Plan. We propose that the new structure presented here be adopted by the IDC as part of normal operating procedures. Results of unit and integration tests are presented in support of this proposal. This enhancement will be applied throughout the entire IDC pipeline since all applications accessing these travel-times facilities must employ a uniform approach. Specifically, this includes the applications, **ARS**, **DFX**, **EvLoc**, **GAcons**, **GAassoc**, **GA_DBI**, **GAconflict**, **LocSAT**, **StaPro**, **WaveExpert**, **wreq** and **XfkDisplay**. Most unit testing was done with application, **EvLoc**, since it permits calculation of batch event locations independent of any other functionality. A description of conventions used throughout this document is included in Appendix F. This evaluation is measured on IDC release version, 4.0.0.

1. CCB MEMO

1.1 Statement of Objectives

The primary goal of this work was the development and implementation of an incremental approach to solving the source/station-region-dependent travel-time problem. Since a full-blown three-dimensional ray tracing approach was both deemed impractical from a computational standpoint, and knowledge-intensive from an implementation point-of-view, an alternative was sought. We adopted an approach whereby relatively simple knowledge could be exploited immediately, for example, one-dimensional velocity models and bulk static station corrections, while more complex information, both model- and data-based, could be integrated as it became available. The approach has been uniformly implemented throughout the entire set of IDC pipeline applications.

1.2 Summary of Proposed Change

The first section of this report is structured to follow the general requirements of an IDC CCB Memo. Since this report is designed to be fairly comprehensive, this first section can be viewed as a significant sub-set of the larger document. Many implementation details are, consequently, left for the appendices so that the primary focus in the main text outlines the overall goals encouraged by the CCB. The proposed change is to replace the current travel-time handling facilities with a new, significantly expanded, set of facilities contained in PIDC release, v. 4.0.0. These facilities support the following new functionality:

- Station-dependent one-dimensional velocity models
- Bulk static station corrections for travel-time
- Sedimentary velocity for individual station/phase pairs (better elevation corrections)
- Source-specific station corrections (SSSC)
- Phase-dependent travel-time modelling errors
- Path-dependent LR and LQ travel times may now be traced (New library, libLP)
- A single velocity model specification file (VMSF) to manage all these items
- Ability to create synthetic travel-time information using application, **EvLoc**.

These facilities require a substantial restructuring of the directories which store the travel-time tables and corrections; these are delivered as part of this delivery. The default configuration recommended at this time does not apply any new corrections, so the behavior will mimic current operational behavior.

The primary motivation for developing a methodology for applying station- and source-dependent knowledge via separate correction terms is to provide an incremental approach whereby individual contributions can be measured against one another. In most cases, specifying an independent one-dimensional velocity model, a bulk station correction and sedimentary velocity for each station will be sufficient to dramatically improve event location accuracy. If no station-dependent velocity model, bulk station correction or separate sedimentary velocity is specified, then the default travel-time tables will be employed [*e.g.*, IASPEI 91 (Kennett, 1991)]. This is the default mode for this PIDC delivery.

Source-specific station correction terms (SSSC) are defined over a geographical rectangular region discretized by latitude and longitude. In the context of this overall model, the source-dependent station correction terms are essentially the final adjustment in the travel-time computation.

Phase-dependent modelling errors, if addressed properly, should significantly improve the accuracy of our error ellipse estimates and will allow us to separate modelling from measurement error. These modelling errors may be specified by a single value, in a distance-dependent manner, or a distance/depth dependent fashion.

The implementation of path-dependent long-period travel time calculations permits detailed ray tracing along the Earth's surface, sub-divided into areas of distinct phase velocity. The ability to create synthetic travel-time information using **EvLoc** will assist researchers in evaluating many aspects of solution sensitivity related to event location. Details are contained within the appendices. All of this new information is specified within a single velocity model specification file described in Appendix C.

Other significant items delivered as part of this release include:

- A new (and recommended) velocity model data (ops) tree
- New AK135 velocity model (Kennett *et al*., 1996) along with associated phase-dependent data modelling errors and ellipticity correction files
- All magnitude correction tables in their own unique area
- A sample EvLoc(1) par file, since this application has changed significantly

1.3 Expected Benefits

The benefits of this new travel-time handling capability include:

- An actual regionalization capability
- A simple incremental approach to the problem
- An enhanced directory structure for storing travel-time & magnitude correction tables
- Backward compatibility
- Data consistency
- A design amenable to other data types
- Functional library independence

The most significant benefit offered by these new facilities is a regionalization capability to permit travel-time knowledge to be added incrementally as it becomes available. These benefits are described in more detail below.

1.3.1 Regionalization Capability

The most substantial benefit to be derived from this new implementation is a full travel-time regionalization capability. The acceptance of this proposed implementation will allow station-, path- and source-dependent information to be integrated in such a way that each contribution can be exploited as available. This has been a long-term goal of the CMR, Group of Scientific Experts (GSE), Air Force Technical Applications Center (AFTAC), and others for quite some time. This implementation represents the first comprehensive attempt at exploiting all three elements of the travel-time problem simultaneously (*i.e.*, source, path and station information) within a global operation setting. It is designed to be incremental such that simple knowledge may be incorporated quickly while more detailed knowledge may be added as it becomes available. We say comprehensive in the sense that each contribution may be separately parameterized unlike test-site or source-region station-time (SRST) corrections (Veith, 1975) which have been used operationally at AFTAC for very many years, but which are total path corrections.

We refer to this as a capability, since the initial travel-time configuration mimics the current system capabilities (*i.e.*, it uses the IASPEI 1991 travel-time tables only). A separate CCB Memo will be needed when we are ready to recommend incorporation of specific corrections.

1.3.2 Incremental Approach

The new travel-time handling facilities provide a practical representation for the evolutionary nature of new travel-time knowledge. All this knowledge is controlled through a single control file defined as the velocity model specification file (VMSF). Through this single file, users will be able to specify for each station/phase pair a different one-dimensional velocity model, static station correction and sedimentary velocity.

1.3.3 Enhanced Table Directory Structure

Travel-time and magnitude correction tables, as delivered, are now stored in separate directories. Now that the VMSF may access numerous travel-time tables, it has become clear that common magnitude correction tables should be defined in their own unique area. Furthermore, since a greater degree of complexity will be demanded permitting access to multiple sets of travel-time tables, the overall directory structure has been modified. Since this same degree of flexibility is envisioned for handling magnitude correction tables in the future, a parallel structure has been forward modelled for these tables as well. This is only a recommended structure. The IDC is of course free to install these tables anywhere they consider appropriate. See section 1.7, *Recommended Installation Procedure for New Travel-Time Facilities*, for these implementation details, including the delivery of new travel-time tables and sample velocity model specification files, recommended for the operational IDC system.

1.3.4 Backward Compatibility

The initial configuration of these new facilities has been set up to provide backward compatibility by mimicking the current travel-time configuration setup. That is, assuming correct integration of the new travel-time table directory structure, the initial set of travel-time tables will employ the IASPEI 1991 (Kennett, 1991) velocity model only. This has allowed us to directly compare results from the previous operational applications to the applications included with this release.

Furthermore, all applications have had the default setting for incorporating distance-dependent variance weighting turned off. Therefore, the only travel-time error used in weighting the event location sensitivity matrix and determining the resultant error ellipse, is extracted from the ***deltim*** field of the ***arrival*** table. This has been the operational default since IDC inception. The ability to employ independent measurement and modelling error exists, but is initially disabled. When the IDC decides to employ independent measurement and phase-dependent modelling errors, then several elements of the IDC pipeline will need to be re-configured. A separate CCB Memo will be required for this step. What is important to realize here is that all functionality currently exists, from **DFX** through **EvLoc**, to install this enhancement.

1.3.5 Data Consistency

Since all corrections are additive, a new approach was required to ensure data consistency between various correction contributions. A structure enabling users to preserve previously determined source-dependent knowledge was critical. For example, if a new bulk static station correction is specified within the VMSF, how do we ensure that previously determined SSSC knowledge is preserved? The approach adopted here is to duplicate information in the velocity model specification file and SSSC files to ensure data consistency with minimal oversite demand on the end user. In this way, if a difference is noted (*e.g.*, a new bulk correction or sedimentary velocity is introduced to the VMSF), then the SSSC values will be statically adjusted relative to the newly specified information. Thus, previously-determined source-dependent knowledge, as stored in the current SSSC file, can still be exploited. Not only will these differences be adjusted following procedures defined in section E.2, *Preserving Data Consistency*, but a newly created SSSC file will automatically be produced (and written to the system /tmp area). Finally, it should be noted that users may control the fact that the travel-time modelling error should be smaller when an SSSC is employed, and this may be mapped via a single-valued attribute contained within the SSSC file that is multiplied to the phase-dependent modelling error (see section E.3, *Source-Dependent Modelling Errors*).

1.3.6 Amenable to Other Data Types

A vital design consideration of these facilities is that they be amenable to other data types (*e.g*., azimuth, slowness, magnitude). This model may be easily adapted to apply corrections to, say, magnitude, in a very analogous way. This is an important consideration. The travel-time problem could have been solved using exact or approximate ray tracing techniques, yet such an implementation could not be applied to these other data types.

1.3.7 Library Independence

In the previous implementation many magnitude library, libmagn(3), functions were significantly dependent of the locator library, libloc(3). These libraries are now totally independent of one another.

1.3.8 Common Magnitude Interface

Although somewhat misplaced within the context of this document, it is yet worth noting that all IDC applications now use the same set of magnitude routines. In particular, ARS(1), now uses the same set of new C functions as EvLoc(1). This should help ensure consistency of results between these two applications. Previously, **ARS** used the FORTRAN routines originally developed by Keith McLaughlin and later modified at the Center by Hans Israelson. Algorithmically, the C and FORTRAN routines perform the same operations. Functionally, the C routines manage common units of information in a more convenient and practical manner. In particular, the C routines have been forward designed to more easily integrate local magnitudes. Currently, only body-wave, m_b, and surface-wave, M_s, magnitudes are estimated.

1.4 Possible Risks

The primary risks are that the new code may crash or produce bad event locations. Extensive unit testing, described in the next section, gives us a reasonable confidence that this is not likely. Unit and integration testing are the best methods for evaluating and assessing risks of individual applications and their interaction with one another in the IDC pipeline. This section will, therefore, only describe risks in general terms.

The magnitude and breadth of changes to almost all of our mission critical IDC applications involved with this delivery increases the likelihood of hidden problems. This new functionality required over 3000 lines of new C source code to be added to library, libloc(3), alone. The new library, libLP(3), constitutes an additional 1500 lines of new C source code. Many applications required a significant integration effort as well, especially **ARS**. Par and scheme file configuration inconsistencies between applications is the greatest concern. Release notes included with this delivery should help to minimize such mishaps.

Another risk is the fact that this significant enhancement is being integrated into a release which represents the first major IDC delivery in over a year. The breadth and scope of totally unrelated enhancements, bug fixes and general changes being included in this delivery increases the cumulative risks by a significant amount. The validation testing documented here goes a long way towards mitigating these risks.

User unfamiliarity with these new travel-time handling facilities presents another risk. While many internal self-consistency checks are made within this new functionality, the flexibility afforded users can be unintentionally abused. Perhaps the most likely errors in this area will occur when installing new travel-time tables. If they are installed in the wrong place within the directory structure the user might assume they are being applied, when in fact, they are not. While this is not a critical problem, it is a risk. It is best mitigated by careful configuration following the guidelines presented in section 1.7, *Recommended Installation Procedure for New Travel-Time Facilities*.

As with the introduction of any new element into an operational system, situations can arise which may be not have been anticipated in the initial design and later development cycles. However, it is expected that any deficiencies may be readily corrected more quickly by gaining operational experience with all affected applications.

1.5 Schedule and Plan for Implementation

Assuming all applications are correctly installed as part of the current PIDC delivery, v. 4.0.0, then the following steps must be verified in order to ensure correct incorporation of these new travel-time handling and related facilities into the IDC automated pipeline:

1. Install complete *PRJ_DIR/ops/static* directory and all sub-directories.
2. Adjust global par file, ‘shared.par’, to include two global settings. The first, VMSF, should be set as, ‘*PRJ_DIR/ops/static/TT/vmsf/iasp91_only.defs*’. The second, ATTEN_PREFIX, should be set as, ‘*PRJ_DIR/ops/static/MAG/atten/atten*’.
3. Point to VMSF location in par files for applications: **DFX**, **EvLoc**, **GAcons**, **GAassoc**, **GAconflict**, **LocSAT**, **StaPro**, **WaveExpert**, **wreq** and **XfkDisplay**. Set the par file argument, vmodel-spec-file=$(VMSF), for all application, except **WaveExpert** and **XfkDisplay**. These two latter applications should use the par file argument name, vmsf.

4. Point to directory and file prefix name in par files for magnitude correction tables for applications: **EvLoc**, **GAassoc**, **GAconflict** and **StaPro**. Set the par file argument, mag-atten-dir-prefix=$(ATTEN_PREFIX).
5. Configure **ARS** to access travel-time and magnitude correction tables by setting the following two scheme variables: (set! velocity-model-spec-file "*PRJ_DIR/ops/static/TT/vmsf/iasp91_only.defs*") and (define magnitude-corr-directory-prefix "*PRJ_DIR/ops/static/MAG/atten/atten*").
6. Verify installation of new default **EvLoc** and **LocSAT** par files where desired as suggested in delivered release notes. Adjust these par files as necessary for adopted IDC configuration.

This effort should be scheduled as soon as all applications have been successfully installed in the IDC pipeline. This should be correctly configured in the testbed and then just copied to operations.

1.6 Cost and Resources Required for Implementation

It is estimated that integrating the items addressed in the previous section should not take more than one full day. This effort should be undertaken by the appropriate IDC personnel. An additional full day effort will be required by Walter Nagy to ensure par, scheme and ASCII files are correctly integrated into the system.

1.7 Recommended Installation Procedure for New Travel-Time Facilities

The new travel-time and magnitude correction tables have been delivered as: *PRJ_DIR/ops/static*, where *PRJ_DIR* is the chosen IDC project directory area. This structure unambiguously suggests that only truly static data, such as travel-time tables, exist within a directory named, static. No par or scheme files are recommended in this area; in this author's view, these are not static data. All travel-time and magnitude correction tables are contained in the sub-directories, *TT* and *MAG*, respectively. This parallelism is intended to provide high-level directory structure for all future source-dependent knowledge as well (*e.g.*, azimuth, slowness, *etc.*). For now, all magnitude correction (attenuation) information is contained in directory, *PRJ_DIR/ops/static/MAG/atten*. In the future, this directory will be sub-divided similarly to the travel-time tables as defined in the next paragraph.

The directory, *PRJ_DIR/ops/static/TT,* has been sub-divided into the following sub-directories:

- *vmsf*: Contains all VMSFs. For this delivery only use file, 'iasp91_only.defs'
- *LP*: Contains path-dependent long-period, LR and LQ, travel-time tables
- *iasp91*: IASPEI 1991 (Kennett, 1991) travel-time and ellipticity correction tables
- *ak135*: AK135 (Kennett *et al.*, 1996) travel-time and ellipticity correction tables
- *herrin68*: Contains Herrin 1968 travel-time tables
- *jb*: Contains Jeffrey-Bullen (J-B) travel-time tables
- *local*: Contains local travel-time tables (only Fennoscandia [fenno] for now)

- *regional*: Contains regional travel-time tables (empty for now)
- *tectonic*: Contains tectonic-based travel-time tables (empty for now)

Sub-directories, *iasp91* and *ak135*, contain their own distance/depth-dependent ellipticity correction tables (see Appendix C for details) under directory, *PRJ_DIR/ops/static/TT/iasp91/Ellip* and *PRJ_DIR/ops/static/TT/ak135/Ellip*, respectively. The sub-directory, *ak135*, is being delivered as requested by Petr Firbas. Although informally delivered to Petr in the past, we now define an official resting place for these tables. Furthermore, both the iasp91 and ak135 travel-time tables contain a preliminary set of phase-dependent modelling errors which were not included in the previous informal delivery. The last three directories are recommended place holders for future one-dimensional travel-time tables. Sub-directory, *local*, only contains regional branches for the Fennoscandia velocity model. The other two sub-directories are currently empty.

Note that sample SSSC sub-directories exist for travel-time tables, *iasp91* and *ak135*. These `SSSC` files (defined in Appendix E) are the result of our GERESS study (Jenkins *et al.*, 1992) conducted over three years ago and are located in the respective sub-directories, *SSSC*. If the `VMSF` includes a specific entry for any station/phase pair, then the `SSSC` information will be read, if it exists. On the other hand, it will only be included as part of the travel-time calculation if requested by the calling routine.

Finally, it is recommended that you replace the long-period curves for LR and LQ phases contained within sub-directory, *LP*, with those tables developed by Jeff Stevens (S-Cubed, San Diego).

1.8 Known Bugs

During integration testing a bug was discovered in **ARS** which has a convenient workaround. If the user attempts to change the `list-of-phases` during an **ARS** session it is possible that freed memory will be accessed which will cause **ARS** to fail. This can be avoided by simply not attempting to reset the `list-of-phases`. This problem should be limited to **ARS** and has been fixed in San Diego. This fix will be delivered with the next release of **ARS**.

2. TESTING

2.1 Description of Testing

The unit and robustness testing described in this section has been performed in San Diego. Integration testing has been largely evaluated on the testbed at the IDC for release, v. 4.0.0. These tests have been conducted to validate the implemented algorithms and the default configuration. They were conducted to minimize adverse side effects such degraded run-time performance, and event solution instabilities before full installation into the operational system. The following sections describe the testing procedures evaluated on a unit basis where applicable and within the pipeline processing on the IDC testbed where necessary.

2.2 Unit Testing

Unit testing was used to validate proper implementation of the new travel-time handling facilities and to evaluate various sensitivity issues using both synthetic and real data. Except for confirming correct integration of the new facilities into other applications, all formal testing described in this section employed the application, **EvLoc** and, to a lesser extent, a non-deliverable program, **Trv_Time**. Unit testing generally consisted of running and evaluating a variety of simple algorithmic tasks for correctness. Since **EvLoc** employs the same travel-time interfaces that are employed throughout the system, validating these facilities with this application validates this functionality within the rest of the applications.

2.2.1 Validation

Validating the correct implementation of these new travel-time handling facilities is the most fundamental and important test we performed. This validation has been addressed systematically by verifying each element of the travel-time calculation. This was easily accomplished given the incremental approach adopted in the design of these new facilities. Two applications were exploited towards this goal, **EvLoc** and **Trv_Time**. Even though **Trv_Time** is a non-deliverable program, it is quite useful for examining various contributions to the total travel-time calculation. **EvLoc** was employed for validation as well, but more frequently, it was used to evaluate event location sensitivity.

2.2.1.1 Validation of Default Configuration

The intention of this initial implementation is to mimic the current set of IDC travel-time tables (*i.e.*, IASPEI 1991) without introducing any adverse side-effects. This is a validation step in and of itself. Validation was accomplished by comparing the previously released IDC version of **EvLoc** to the new version. For four well-determined sample events (from ADSN data, June 23, 1991; number of defining phases > 10), the final event locations agreed to at least four significant figures beyond the decimal point (in degrees) in all cases. In the worst case this amounted to a difference of less than 0.02 km. Given that many variables have been promoted from floats to doubles and a slightly higher resolution set of ellipticity corrections have now been employed, some variation is to be expected. But clearly the resultant differences are not significant. This test verifies a number of features are being correctly applied. These include: the correct one-dimensional travel-time table values are applied as intended; the same elevation correction is used; and no unexpected artifact is being introduced. This test demonstrates that these new travel-time handling facilities can be configured to mimic the current IDC location behavior.

2.2.1.2 'Simulated' Source-Specific Station Corrections

As a test of the application of source-specific station corrections and bulk static station corrections, we computed simulated SSSCs to correct a set of global travel-time tables (model AK135 of Kennett *et al.*, 1996) to a set of regional velocity models for the Fennoscandia region. We then compared results obtained using both sets of travel-time tables.

Trv_Time was used to examine the individual contributions that, in sum, yield our total travel time. **Trv_Time** computes and displays all travel-time contributions for a list of stations and phases given an input event location. The event location may be directly specified by the user interactively or through an origin-based database query. The list of stations to be displayed is extracted from the ***site*** table.

The events that were located were obtained from the IDC ground-truth database account, gtdb/gtdb. The regional phase data compiled in this account was meticulously constructed by Grant and Coyne (1992) for well-determined earthquakes and quarry blasts located in the Fennoscandia region. For the purposes of this testing only Pn, Pg, Sn and Lg phases were examined.

We used this data to compare two velocity models. The first model (VM_{fenno_only}) uses only the Fennoscandia velocity model as currently implemented at NORSAR. The second model (VM_{mimic_fenno}) is structured to mimic the Fennoscandia model by using the one-dimensional AK135 (Kennett *et al*., 1996) velocity model, bulk station corrections and 'simulated' SSSC files for most, but not all, stations listed in Table 1. The last column of Table 1 indicates whether or not a 'simulated' SSSC file was constructed. If not, the one-dimensional Fennoscandia model was assumed. Table 1 also displays the number of associations (***nass***) and number of time defining associations (***ndef***) for each station. We say, 'simulated' SSSC files, since these were constructed only for the purpose of mimicing the one-dimensional Fennoscandia model with source-dependent travel-time corrections. Note that the stations chosen for employing these SSSCs are those with the largest number of time defining (***ndef***) associations. This will become more relevant in the discussion of sensitivity tests. Bulk station corrections were also integrated into the computation for stations, FIA0 and KSP. Since only regional phases were investigated here, only stations at distances less than 20 arc degrees were evaluated in this test. The VMSF used to build VM_{mimic_fenno} is included in Appendix G.

Tables 2 and 3 summarize the travel-time results from velocity models, VM_{fenno_only} and VM_{mimic_fenno}, respectively. These results are sorted by distance relative to the event location and display the individual contributions to the total travel time including, the one-dimensional velocity model and associated travel-time table value, ellipticity correction, elevation correction, bulk station correction and SSSC. All measures are in seconds and no extrapolation of travel times were permitted. The most important observation to note from these results is the near equivalency of the total travel times displayed. While the individual one-dimensional models yield travel time corrections in excess of 11 sec at station, KSP, for phase, Sn, the largest total travel time difference is only 0.14 sec. This latter variation is due to the discretization of the SSSC information itself. This test demonstrates that these new travel-time facilities have been properly implemented. As a note, the small differences sometimes noted in the elevation corrections is the result of differences in slowness relative to the two one-dimensional velocity models employed.

Table 1: Summary of Station, Association and Velocity Model Information

STATION	Latitude (deg)	Longitude (deg)	Elevation (km)	nass	ndef	Mock SSSC File Used
APA0	67.6061	32.9931	0.200	190	112	Yes
APZ9	67.5686	33.4050	0.200	326	180	Yes
ARA0	69.5349	25.5048	0.403	820	555	Yes
ENN	56.7670	5.9230	0.120	2	1	No
FIA0	61.4436	26.0771	0.155	230	166	Yes
FIA1	61.4444	26.0793	0.138	5	5	No
GEC2	48.8451	13.7016	1.132	650	401	Yes
GRA1	49.6920	11.2220	0.500	28	25	Yes
HFS	60.1335	13.6836	0.265	6	5	No
KAF	62.1127	26.3062	0.205	1	1	No
KSP	50.8433	16.2933	0.380	128	87	Yes
LOR	47.2667	3.8514	5.200	8	7	No
NRA0	60.7353	11.5414	0.302	330	234	Yes
OSS	46.6896	10.1331	1.600	11	7	No
SQTA	47.2205	11.2087	1.307	9	9	No
VAF	63.0422	22.6715	0.550	1	1	No
VRAC	49.3087	16.5954	0.480	47	38	Yes
YKA	62.4932	-114.6053	0.197	5	5	No

Table 2: Travel-Time Information using Fennoscandia Model Only, VM_{fenno_only}

ORID	EVID	Latitude (deg)	Longitude (deg)	Depth (km)
1148	2063	67.6702 N	33.7285 E	0.00

```
AP29   Lat:  67.5686  Lon:   33.4050  Elev: 0.2000  Dist:    0.1605  SEaz:  50.361
  Velocity Model Phase     Total TT = Table     + Ellip +  Elev +  Bulk + SRC_CORR
  -------------- --------  --------   ---------  -----    -----    -----  --------
  Fennoscandia   Pg           2.891 =    2.879 + 0.000 + 0.012 + 0.000 + 0.000
  Fennoscandia   Lg           5.048 =    5.028 + 0.000 + 0.020 + 0.000 + 0.000

APA0   Lat:  67.6061  Lon:   32.9931  Elev: 0.2000  Dist:    0.2887  SEaz:  76.769
  Velocity Model Phase     Total TT = Table     + Ellip +  Elev +  Bulk + SRC_CORR
  -------------- --------  --------   ---------  -----    -----    -----  --------
  Fennoscandia   Pg           5.190 =    5.177 + 0.000 + 0.012 + 0.000 + 0.000
  Fennoscandia   Lg           9.062 =    9.042 + 0.000 + 0.020 + 0.000 + 0.000

ARA0   Lat:  69.5349  Lon:   25.5058  Elev: 0.4030  Dist:    3.5477  SEaz: 117.970
  Velocity Model Phase     Total TT = Table     + Ellip +  Elev +  Bulk + SRC_CORR
  -------------- --------  --------   ---------  -----    -----    -----  --------
  Fennoscandia   Pn          56.088 =   56.145 - 0.106 + 0.049 + 0.000 + 0.000
  Fennoscandia   Pg          63.652 =   63.628 + 0.000 + 0.025 + 0.000 + 0.000
  Fennoscandia   Sn          98.126 =   98.227 - 0.185 + 0.084 + 0.000 + 0.000
  Fennoscandia   Lg         111.163 =  111.124 + 0.000 + 0.040 + 0.000 + 0.000

KAF    Lat:  62.1127  Lon:   26.3062  Elev: 0.2050  Dist:    6.4073  SEaz:  26.253
  Velocity Model Phase     Total TT = Table     + Ellip +  Elev +  Bulk + SRC_CORR
  -------------- --------  --------   ---------  -----    -----    -----  --------
  Fennoscandia   Pn          94.646 =   94.779 - 0.158 + 0.025 + 0.000 + 0.000
  Fennoscandia   Pg         114.926 =  114.913 + 0.000 + 0.012 + 0.000 + 0.000
  Fennoscandia   Sn         165.933 =  166.168 - 0.278 + 0.043 + 0.000 + 0.000
  Fennoscandia   Lg         200.714 =  200.693 + 0.000 + 0.020 + 0.000 + 0.000

FIA0   Lat:  61.4436  Lon:   26.0771  Elev: 0.1550  Dist:    7.0609  SEaz:  24.450
  Velocity Model Phase     Total TT = Table     + Ellip +  Elev +  Bulk + SRC_CORR
  -------------- --------  --------   ---------  -----    -----    -----  --------
  Fennoscandia   Pn         103.459 =  103.610 - 0.170 + 0.019 + 0.000 + 0.000
  Fennoscandia   Pg         126.645 =  126.635 + 0.000 + 0.009 + 0.000 + 0.000
  Fennoscandia   Sn         181.430 =  181.697 - 0.300 + 0.032 + 0.000 + 0.000
  Fennoscandia   Lg         221.181 =  221.166 + 0.000 + 0.015 + 0.000 + 0.000

HFS    Lat:  60.1335  Lon:   13.6836  Elev: 0.2650  Dist:   11.5630  SEaz:  40.800
  Velocity Model Phase     Total TT = Table     + Ellip +  Elev +  Bulk + SRC_CORR
  -------------- --------  --------   ---------  -----    -----    -----  --------
  Fennoscandia   Pn         164.208 =  164.437 - 0.262 + 0.032 + 0.000 + 0.000
  Fennoscandia   Pg         207.395 =  207.378 + 0.000 + 0.016 + 0.000 + 0.000
  Fennoscandia   Sn         288.256 =  288.665 - 0.464 + 0.055 + 0.000 + 0.000
  Fennoscandia   Lg         362.208 =  362.182 + 0.000 + 0.026 + 0.000 + 0.000

NRA0   Lat:  60.7353  Lon:   11.5414  Elev: 0.3020  Dist:   11.8410  SEaz:  44.688
  Velocity Model Phase     Total TT = Table     + Ellip +  Elev +  Bulk + SRC_CORR
  -------------- --------  --------   ---------  -----    -----    -----  --------
  Fennoscandia   Pn         167.961 =  168.194 - 0.270 + 0.037 + 0.000 + 0.000
  Fennoscandia   Pg         212.383 =  212.364 + 0.000 + 0.018 + 0.000 + 0.000
  Fennoscandia   Sn         294.854 =  295.270 - 0.479 + 0.063 + 0.000 + 0.000
  Fennoscandia   Lg         370.919 =  370.890 + 0.000 + 0.030 + 0.000 + 0.000

KSP    Lat:  50.8433  Lon:   16.2933  Elev: 0.3800  Dist:   18.9530  SEaz:  20.641
  Velocity Model Phase     Total TT = Table     + Ellip +  Elev +  Bulk + SRC_CORR
  -------------- --------  --------   ---------  -----    -----    -----  --------
  Fennoscandia   Pn         263.984 =  264.283 - 0.345 + 0.046 + 0.000 + 0.000
  Fennoscandia   Sn         463.702 =  464.248 - 0.626 + 0.079 + 0.000 + 0.000
  Fennoscandia   Lg         593.693 =  593.656 + 0.000 + 0.037 + 0.000 + 0.000
```

Table 3: Travel-Time Information using AK135 Model + Corrections, VM_{mimic_fenno}

ORID	EVID	Latitude (deg)	Longitude (deg)	Depth (km)
1148	2063	67.6702 N	33.7285 E	0.00

```
APZ9   Lat:  67.5686  Lon:   33.4050  Elev: 0.2000  Dist:   0.1605  SEaz:  50.361
  Velocity Model Phase    Total TT = Table    + Ellip +  Elev +  Bulk + SRC_CORR
  -------------- -------- --------   --------   -----    -----    -----   --------
  ak135          Pg          2.879 =    2.926 + 0.000 + 0.011 + 0.000 - 0.058SS
  ak135          Lg          5.032 =    5.100 + 0.000 + 0.017 + 0.000 - 0.085SS

APA0   Lat:  67.6061  Lon:   32.9931  Elev: 0.2000  Dist:   0.2887  SEaz:  76.769
  Velocity Model Phase    Total TT = Table    + Ellip +  Elev +  Bulk + SRC_CORR
  -------------- -------- --------   --------   -----    -----    -----   --------
  ak135          Pg          5.186 =    5.262 + 0.000 + 0.011 + 0.000 - 0.087SS
  ak135          Lg          9.058 =    9.171 + 0.000 + 0.017 + 0.000 - 0.131SS

ARA0   Lat:  69.5349  Lon:   25.5058  Elev: 0.4030  Dist:   3.5477  SEaz: 117.970
  Velocity Model Phase    Total TT = Table    + Ellip +  Elev +  Bulk + SRC_CORR
  -------------- -------- --------   --------   -----    -----    -----   --------
  ak135          Pn         56.083 =   56.310 - 0.106 + 0.048 + 0.000 - 0.169SS
  ak135          Pg         63.656 =   64.670 + 0.000 + 0.022 + 0.000 - 1.036SS
  ak135          Sn         98.126 =   98.945 - 0.185 + 0.080 + 0.000 - 0.715SS
  ak135          Lg        111.160 =  112.711 + 0.000 + 0.035 + 0.000 - 1.586SS

KAF    Lat:  62.1127  Lon:   26.3062  Elev: 0.2050  Dist:   6.4073  SEaz:  26.253
  Velocity Model Phase    Total TT = Table    + Ellip +  Elev +  Bulk + SRC_CORR
  -------------- -------- --------   --------   -----    -----    -----   --------
  Fennoscandia   Pn         94.646 =   94.779 - 0.158 + 0.025 + 0.000 + 0.000
  Fennoscandia   Pg        114.926 =  114.913 + 0.000 + 0.012 + 0.000 + 0.000
  Fennoscandia   Sn        165.933 =  166.168 - 0.278 + 0.043 + 0.000 + 0.000
  Fennoscandia   Lg        200.714 =  200.693 + 0.000 + 0.020 + 0.000 + 0.000

FIA0   Lat:  61.4436  Lon:   26.0771  Elev: 0.1550  Dist:   7.0609  SEaz:  24.450
  Velocity Model Phase    Total TT = Table    + Ellip +  Elev +  Bulk + SRC_CORR
  -------------- -------- --------   --------   -----    -----    -----   --------
  ak135          Pn        103.460 =  104.584 - 0.170 + 0.019 - 0.300 - 0.673SS
  ak135          Pg        126.645 =  128.711 + 0.000 + 0.008 - 0.500 - 1.575SS
  ak135          Sn        181.429 =  185.544 - 0.300 + 0.031 - 0.700 - 3.146SS
  ak135          Lg        221.178 =  224.325 + 0.000 + 0.013 - 0.600 - 2.561SS

HFS    Lat:  60.1335  Lon:   13.6836  Elev: 0.2650  Dist:  11.5630  SEaz:  40.800
  Velocity Model Phase    Total TT = Table    + Ellip +  Elev +  Bulk + SRC_CORR
  -------------- -------- --------   --------   -----    -----    -----   --------
  Fennoscandia   Pn        164.208 =  164.437 - 0.262 + 0.032 + 0.000 + 0.000
  Fennoscandia   Pg        207.395 =  207.378 + 0.000 + 0.016 + 0.000 + 0.000
  Fennoscandia   Sn        288.257 =  288.665 - 0.464 + 0.055 + 0.000 + 0.000
  Fennoscandia   Lg        362.209 =  362.182 + 0.000 + 0.026 + 0.000 + 0.000

NRA0   Lat:  60.7353  Lon:   11.5414  Elev: 0.3020  Dist:  11.8410  SEaz:  44.688
  Velocity Model Phase    Total TT = Table    + Ellip +  Elev +  Bulk + SRC_CORR
  -------------- -------- --------   --------   -----    -----    -----   --------
  ak135          Pn        167.959 =  170.098 - 0.270 + 0.036 + 0.000 - 1.905SS
  ak135          Pg        212.381 =  215.846 + 0.000 + 0.016 + 0.000 - 3.481SS
  ak135          Sn        294.856 =  302.934 - 0.479 + 0.061 + 0.000 - 7.660SS
  ak135          Lg        370.916 =  376.188 + 0.000 + 0.026 + 0.000 - 5.298SS

KSP    Lat:  50.8433  Lon:   16.2933  Elev: 0.3800  Dist:  18.9530  SEaz:  20.641
  Velocity Model Phase    Total TT = Table    + Ellip +  Elev +  Bulk + SRC_CORR
  -------------- -------- --------   --------   -----    -----    -----   --------
  ak135          Pn        263.956 =  263.215 - 0.345 + 0.051 - 0.400 + 1.435SS
  ak135          Sn        463.846 =  475.963 - 0.626 + 0.078 - 1.000 -10.569SS
  ak135          Lg        593.801 =  602.137 + 0.000 + 0.033 - 1.000 - 7.369SS
```

2.2.2 Synthetic Data Tests

All applications included in the PIDC v. 4.0.0 release use the same travel-time handling facilities, so sensitivity testing evaluated using **EvLoc** is sufficient to demonstrate characteristics of the new travel-time handling functionality. We refer to this evaluation as sensitivity testing, since we can essentially measure how sensitive our new implementation is to controlled (synthetic) data. Two synthetic data sets are explored in some detail within this section. The first is an idealized synthetic data set where the theoretical arrival time, azimuth and slowness is equal to the observed arrival time, azimuth and slowness. A second synthetic data set is created employing random perturbations about a Gaussian-distributed set of observed arrival times, azimuths and slownesses. **EvLoc** is employed since it's the only application capable of reading event information directly from the database, calculating theoreticals, and outputting these synthetic results back out to the database in the form of new ***origin***, ***assoc*** and ***arrival*** tables.

This testing further validates the new travel-time handling implementation and clearly demonstrates the practical value of examining synthetic data. These tests will demonstrate that a laterally variable heterogeneous Earth model may be adequately represented by a simple station-dependent one-dimensional velocity model, bulk station correction, locally-defined sedimentary velocity and source-dependent corrections, namely SSSCs in this document. While event locations employing the Fennoscandia vs. AK135 velocity models differ, on average, by 7.75 km., the distance difference between the Fennoscandia velocity model and one designed to mimic it (*i.e.,* the AK135 model plus bulk station corrections and SSSCs) is only 0.17 km.

By creating synthetic data from a real event data set we introduce the concept of phase-dependent modelling errors being totally independent from the measurement error. By assuming distance-dependent travel-time modelling errors for our four regional phases, we hope to demonstrate the value of this new capability. It will be shown that by adding random perturbations to our synthetic data, based on the actual errors in our real arrival data set, that 90% of the entire set of re-located events are contained within the final 90% confidence ellipses.

Finally, we will present results which suggests that our current procedural rules for diassociating arrivals or making them non-defining results in an unintended data censuring. This censuring is a function of the number of defining data assigned to the given event and suggests we may wish to re-evaluate our current guidelines.

2.2.2.1 Idealized Synthetic Data

As a first step in this evaluation, **EvLoc** was used to compute idealized synthetic (theoretical) arrival times, azimuths and slownesses for 288 events included in the gtdb/gtdb database account relative to the one-dimensional Fennoscandia velocity model (VM_{fenno_only}). Since all associations were originally set time, azimuth and slowness non-defining in this account, it was necessary to adjust their settings to simulate normal IDC conditions. For the purposes of this testing all Pn, Pg, Sn and Lg phases were made time defining. Valid azimuths and slownesses were also made defining. The exception was for Pg and Lg slownesses which were left non-defining.

Additionally, now that measurement and modelling error can be handled as independent contributions to the total travel-time error, we modified the ***deltim*** attribute of the ***arrival*** table to reflect only the measurement error. This measurement error was defined as a function of the signal-to-noise ratio (SNR) such that ***deltim*** was populated as follows:

- SNR ≤ 5.0: ***deltim*** = 1.0 sec.
- SNR ≥ 50.0: ***deltim*** = 0.1 sec.
- 50.0 > SNR > 5.0: ***deltim*** = 1.0 - (0.9 * $\log_{10}$ (SNR/5.0))

The phase-dependent modelling error, as extracted from the travel-time tables (see Appendix C), is combined into the total error along with ***deltim*** as a combined RMS measure (*i.e.*, the total variance (σ^2_{tot}) = measurement variance (σ^2_{meas}) + modelling variance (σ^2_{model})). The inverse of the total error assigned to the arrival time is the weight used for that datum for the location process.

All 288 events relocated in 4 iterations (the minimum number required) employing VM_{fenno_only} to within 0.001 km. of synthetically-produced (true) input event location. This step validates that the synthetics were correctly computed. The next step was to relocate all these events assuming the AK135 velocity model (VM_{ak135_only}) of Kennett *et. al* (1996), thereby mimicking the IDC configuration prior to PIDC release v. 4.0.0. In this case, the average distance difference, relative to the synthetic event locations, is 7.75 km., the largest being 72.6 km.

We would hope that by employing the velocity model, VM_{mimic_fenno}, as we did in section 2.2.1.2, *'Simulated' Source-Specific Station Corrections*, we could use the one-dimensional AK135 velocity model plus bulk station corrections and SSSCs to sufficiently mimic the Fennoscandia model. Relocating all 288 events yielded an average event location difference of 0.17 km. between the synthetic event locations and the final event locations determined using VM_{mimic_fenno}. The largest distance difference was 2.35 km., while the median difference was only 0.04 km. These results indicate that we can indeed mimic VM_{fenno_only} with VM_{mimic_fenno} with little loss of accuracy. This further validates that this new implementation correctly applies all travel-time contributions as intended. Additionally, it also validates its correct application within the location process itself.

2.2.2.2 Synthetics with Gaussian Noise Added

To further assess our new travel-time handling capability, especially with regard to separating measurement and modelling error, a new synthetic data set was created with random perturbations following a Gaussian distribution about the arrival times, azimuths and slownesses. The same 288 events which were assessed in the previous section were re-evaluated with these random perturbations applied.

Random perturbations were added to the theoretical azimuth and slowness measures, on a per arrival basis, about one standard deviation of the ***arrival*** table attributes, ***delaz*** and ***delslo***, respectively, following a Gaussian distribution. For the theoretical arrival times, these perturbations are distributed about one standard deviation of the combined RMS measure constituting the measurement error, ***deltim***, and the modelling error. The modelling error was extracted from the phase-dependent travel-time tables as either a single static value, from a distance-dependent table or from a distance/depth-dependent table (see Appendix C for details). In this way the measurement error is strictly a function of the SNR, while the modelling error only encompasses our lack of knowledge about the true travel-time path averaged over the one-dimensional model assigned to the given phase type. If an SSSC is to be added to the travel-time calculation, then a locally-specified modelling error factor (**me_factor**) is multiplied to the phase-dependent modelling error extracted directly from the travel-time table. We would expect this factor to generally vary from 0.0 to 1.0, approaching 0.0 as more and more knowledge is mapped into the station/phase-dependent SSSC grid. For this study only distance-dependent travel-time modelling error was explored with **me_factor** set to 1.0 to mimic modelling errors assumed in VM_{fenno_only}. The distance-dependent travel-time error (sec.) for the four regional phases employed in this study is shown in Table 4. Intermediate values are determined via simple linear interpolation.

Table 4: Phase-Dependent Travel-Time Modelling Error for Regional Phases

Distance (deg)	P_n T-T Error (sec.)	S_n T-T Error (sec.)	Distance (deg)	P_g T-T Error (sec.)	L_g T-T Error (sec.)
0.0	0.1	0.1	0.0	0.1	0.1
16.0	3.0	5.0	20.0	10.0	20.0
18.0	0.7	1.2			
21.0	0.7	1.2			

The distribution of arrival times, azimuths and slownesses about their theoretically computed values for the four regional phases is summarized in Table 5. Note that some very large perturbations ($\delta\tau$, $\delta\zeta$ and $\delta\gamma$; for arrival time, azimuth and slowness, respectively) are introduced into our new synthetic data set, although their mean is meant to coincide with the assumption of no systematic bias in the constructed model (*i.e.*, all averages tend toward 0.0). The resultant phase-dependent standard deviations match the distance-weighted modelling error distribution as would be expected. Table 6 sub-divides the arrival time results of Table 5 into specific station/phase contributions. It is worth noting that all time perturbations greater than 12 sec. occur only for L_g phases at relatively far regional distances (> 7°) where the phase-dependent modelling error is large (*i.e*., 1.0 sec. for every 1°; note stations FIA0 and NRA0 in Table 6). From a cursory inspection of these results one might be inclined to suggest that no trained analyst would ever associate an arrival with a 20 sec. anomaly, we will show below that for small ***ndef*** events this can and does occur resulting in a new event location with small travel-time residuals (and sometimes small error ellipses).

Table 5: Distribution of Random Theoretical Arrival Time, Azimuth and Slowness Perturbations ($\delta\tau$; $\delta\zeta$; $\delta\gamma$) for Defining Regional Phases at All Stations

PHASE	Arrival Time Perturbations ($\delta\tau$; sec)				Azimuth Perturbations ($\delta\zeta$; deg)				Slowness Perturbations ($\delta\gamma$; sec/deg)			
	#	$\delta\tau_{max}$	$\overline{\delta\tau}$	σ	#	$\delta\zeta_{max}$	$\overline{\delta\zeta}$	σ	#	$\delta\gamma_{max}$	$\overline{\delta\gamma}$	σ
P_n	445	4.60	-0.01	1.45	397	72.93	0.89	11.05	397	41.40	-0.13	4.12
P_g	371	6.31	0.04	1.52	252	30.27	0.02	8.26	N/A			
S_n	322	8.74	0.15	2.23	222	65.98	0.13	7.48	222	18.80	-0.07	4.29
L_g	607	27.69	-0.03	3.35	436	57.80	-0.30	9.44	N/A			

The 288 events were relocated using the previously described velocity models, VM_{fenno_only}, VM_{mimic_fenno}, and VM_{ak135_only}. For all three velocity models, all but four solutions converged leaving a subset of 284 events. The four events that failed all occur at the Kirovsk mine where station, APZ9, lies at a distance of less than 20 km. As it turns out, the random arrival time perturbations have caused L_g to occur either before or within 0.2 sec. of P_g in all four cases. This behavior created a very undesirable condition for the locator thereby degrading the likelihood of convergence.

The average distance difference between the synthetic (true) event locations and the relocated events using VM_{fenno_only} with random perturbation applied to arrival time, azimuth and slowness data is $\overline{\delta\Delta}_{syn\_fenno}$ = 18.2 km. (The median distance difference is $\delta\Delta_{syn\_fenno}$ = 10.8 km. and the largest distance difference is $\delta\Delta_{syn\_fenno}$ = 249 km). The average distance differences are only slightly higher when comparing the synthetic event locations with locations determined using velocity models, VM_{mimic_fenno} ($\delta\Delta_{syn\_mimic}$ = 18.3 km.) and VM_{ak135_only} ($\delta\Delta_{syn\_ak135}$ = 20.8 km). Thus, given our present assumptions about measurement and modelling error, with a highly-reviewed data set assuming no systematic bias in the average velocity model, we can consider, on average, an ~18 km. absolute error to be achievable. This maps to an error ellipse area (Θ) of approximately 4,000 km^2. Perhaps a more meaningful measure of this error is the median distance difference of approximately 11 km. In this case, the resultant error ellipse error would be about 1,500 km^2. In either case, the final error ellipse exceeds the target of 1,000 km^2 set by the GSE. Indeed, as we would expect, approximately 90% of the resultant error ellipses contain the true event locations.

Table 6: Distribution of Arrival Time Perturbations ($\delta\tau$) for Each Station/Phase Pair

STATION	PHASE	Num	$\delta\tau_{min}$	$\delta\tau_{max}$	$\overline{\delta\tau}$	σ
APA0	Lg	52	-2.01	2.24	-0.04	0.79
	Pg	54	-2.05	0.89	-0.07	0.60
APZ9	Lg	101	-2.44	2.04	-0.08	1.05
	Pg	69	-2.04	2.28	-0.04	1.08
ARA0	Lg	143	-9.40	10.03	0.15	3.83
	Pg	81	-6.31	5.26	0.32	2.01
	Pn	139	-2.46	2.00	-0.02	0.92
	Sn	141	-3.96	4.14	0.08	1.57
ENN	Pn	1	-0.16	-0.16	-0.16	0.00
FIA0	Lg	55	-12.72	15.25	0.20	6.20
	Pn	61	-4.15	3.57	-0.02	1.81
	Sn	47	-7.92	5.75	-0.22	2.75
FIA1	Lg	1	10.98	10.98	10.98	0.00
	Sn	4	-2.61	5.07	0.56	3.47
GEC2	Lg	130	-9.89	11.60	-0.51	3.58
	Pg	117	-5.30	5.76	0.03	1.76
	Pn	94	-2.76	3.59	0.18	1.17
	Sn	48	-3.48	2.65	0.19	1.37
GRA1	Lg	12	-2.35	3.24	0.17	1.66
	Pg	7	-1.20	1.83	0.16	1.10
	Pn	6	-1.88	0.62	-0.50	1.06
HFS	Pn	5	-3.69	4.33	0.87	3.06
KAF	Pn	1	-0.14	-0.14	-0.14	0.00
KSP	Lg	40	-9.61	6.50	-0.41	2.46
	Pg	32	-4.97	3.00	-0.25	1.21
	Pn	13	-1.58	3.12	0.28	1.28
	Sn	2	3.03	5.89	4.46	2.02

Table 6: Distribution of Arrival Time Perturbations ($\delta\tau$) for Each Station/Phase Pair

STATION	PHASE	Num	$\delta\tau_{min}$	$\delta\tau_{max}$	$\overline{\delta\tau}$	σ
LOR	Lg	3	-7.46	9.30	0.46	8.42
	Pn	3	0.32	2.65	1.24	1.24
	Sn	1	-0.54	-0.54	-0.54	0.00
NRA0	Lg	47	-21.42	27.69	0.40	10.19
	Pg	1	0.72	0.72	0.72	0.00
	Pn	103	-4.60	3.94	-0.21	1.91
	Sn	76	-8.43	8.74	0.36	3.10
OSS	Lg	3	-3.58	0.82	-1.50	2.21
	Pn	4	-0.65	0.28	-0.21	0.43
SQTA	Lg	2	-0.93	3.17	1.12	2.90
	Pn	5	-1.47	3.01	-0.04	1.78
	Sn	2	-0.18	1.83	0.82	1.42
VAF	Pn	1	1.69	1.69	1.69	0.00
VRAC	Lg	18	-6.48	5.05	0.58	2.63
	Pg	10	-1.91	1.66	-0.05	1.05
	Pn	9	-1.91	2.20	-0.48	1.24
	Sn	1	0.61	0.61	0.61	0.00

Upon closer inspection we see a need to qualify this statement. Table 7 displays the average and median 90th percentile error ellipse information vs. the location error for events located with velocity model, VM_{fenno_only}, relative to the true synthetic event locations, grouped by the number of time defining phases. (The error ellipse calculated here assumes exact *a priori* knowledge). What is immediately obvious from this table is that events with fewer than seven time defining phases clearly fail to meet our stated goal of error ellipses having an area of < 1,000 km^2. In fact, events with four or fewer time defining phases fall woefully short of our target. 44% of the complete re-located event set meet this stated criteria, but for ***ndef*** < 7 events this number drops to 16%. And all these events are located with the depth element constrained. This means we are going to have to develop significantly better velocity models that can bring down our modelling errors quite substantially if we hope to reach the goals stated by the GSE.

Table 7: Relationship between ndef, 90th percentile error ellipse and location error

ndef	**Num**	$\overline{\textbf{smajax}}$ **(km)**	$\overline{\textbf{sminax}}$ **(km)**	$\overline{\Theta}$ **(km^2)**	**Median** Θ **(km^2)**	$\overline{\delta\Delta}_{syn\_fenno}$ **(km)**
2	40	80.78	50.10	16,659	5,390	38.19
3	28	106.26	50.75	22,428	10,431	40.40
4	53	40.13	24.20	3,214	2,963	19.12
5	24	36.45	16.01	1,968	1,779	14.68
6	23	28.01	12.90	1,283	1,373	12.78
7	25	26.41	12.32	1,121	969	11.95
8	32	14.75	7.07	415	173	7.25
9	14	16.55	9.02	584	293	6.29
10	20	19.76	9.37	613	690	6.17
11	6	11.61	7.21	298	278	5.33
12	2	11.47	7.56	296	296	4.56
13	4	12.89	7.50	317	367	2.63
14	8	11.42	5.99	220	191	4.54
15	4	12.56	7.83	309	315	5.84
18	1	17.56	8.16	450	450	5.11

Table 8 points to a serious issue related to the interpretation of resultant *a posteriori* travel-time residuals (***timeres***) determined for low ***ndef*** events. Again, with more than seven defining phases we find that the travel-time residuals (***timeres***) are of comparable size with the distribution of average arrival time perturbations ($\overline{\delta\tau}$), as would be expected. But this behavior breaks down rapidly for lower ***ndef*** events. At the extreme end, travel-time perturbations as large as 6.79 sec. are reduced to zero for ***ndef*** = 2 events. In this case, we have no hope of objectively removing outliers like we would under normal operating procedures for large ***ndef*** events. We simply map these uncertainties into the event location which has a very small RMS travel-time residual (***sdobs***). But even for ***ndef*** 3 and 4 events, arrival-time perturbations > 10 sec. very often dramatically result in travel-time residuals that are small. This effect is reduced for ***ndef*** 5, 6 and 7 events, but still substantial.

What these results imply is that by making phases with large residuals non-defining, or disassociating them altogether, for large ***ndef*** events, we are, in effect, censuring our data differently for large and small events. This has far-reaching implications when it comes to our final interpretations. How do we evaluate such small ***ndef*** events and should we?

Table 8: Relationship between ndef, Arrival Time Perturbation ($\delta\tau$) and Final Travel-Time Residual

ndef	Num	$\overline{\text{sdobs}}$ (sec)	$\overline{\text{smajax}}$ (km)	$\overline{\text{sminax}}$ (km)	$\overline{\text{distance}}$ (deg)	$\delta\tau_{max}$ (sec)	$\overline{\delta\tau}$ (sec)	$\overline{\text{timeres}}$ (sec)
2	80	0.001	80.79	50.10	3.27	6.791	1.728	0.001
3	84	1.048	106.26	50.75	4.50	11.603	1.873	0.876
4	212	1.339	40.13	24.20	3.38	10.028	1.574	0.989
5	120	1.784	36.45	16.01	4.70	9.399	1.659	1.296
6	138	1.967	28.00	12.90	4.15	18.634	1.867	1.385
7	175	2.252	26.41	12.32	4.71	15.246	1.912	1.530
8	256	2.206	14.75	7.07	3.11	13.775	1.558	1.448
9	126	2.956	16.55	9.02	4.71	21.415	2.043	1.910
10	200	2.831	19.76	9.37	5.42	16.650	1.994	1.901
11	66	2.865	11.61	7.21	4.42	10.976	2.009	1.973
12	24	3.126	11.47	7.56	4.52	12.721	1.996	1.713
13	52	4.994	12.89	7.50	4.71	27.690	2.651	2.659
14	112	2.439	11.42	5.99	4.92	8.688	1.716	1.668
15	60	2.048	12.56	7.83	5.39	9.609	1.489	1.402
18	18	3.064	17.56	8.16	6.59	9.298	2.117	2.002

The final stage of this validation testing is analogous to the event relocation process undertaken with our perfect synthetic data set from section 2.2.2.1, *Idealized Synthetic Data*, to evaluate solution sensitivity. In particular, how well do our new travel-time handling facilities map event locations using VM_{fenno_only} to VM_{mimic_fenno} in the presence of Gaussian perturbations? This was evaluated by again comparing the average distance differences (km.) between events located using velocity model, VM_{fenno_only}, and the two velocity models, VM_{mimic_fenno} and VM_{ak135_only}. While the average distance differences between the synthetics and all three velocity models are more than 18 km., the average distance difference between VM_{fenno_only} and VM_{mimic_fenno} is only 0.53 km. Further, the median distance difference is only 0.15 km. with the largest difference being 47.2 km. At the same time, the average distance difference between VM_{fenno_only} and VM_{ak135_only} is almost exactly the same as the perfect synthetic case (average: 7.59 km.; largest: 82.5 km.). This test validates, that even in the presence of Gaussian noise, we can satisfactorily define a model whereby heterogeneous velocity information can be mapped using our new travel-time handling facilities.

2.2.3 Robustness Testing

A performed in San Diego as part of PIDC v. 4.0.0 release process. Some generalities observed during this evaluation are discussed here. This testing is differentiated from integration testing which was conducted on the IDC testbed. Many IDC mission critical elements were tested. Applications not included in this testing section were tested during the San Diego release process, but were not the best candidates for measuring robustness.

2.2.3.1 The Global Association (GA) Sub-System

As a test of robustness of these new travel-time handling facilities, five days of IDC data were processed by the GA sub-system. This system is integrated as part of the IDC pipeline and invokes applications, **GAassoc**, **GAconflict** and **GA_DBI**. It was configured to use the default IASPEI 1991 configuration as prescribed here for the initial IDC installation. Inspection of the output log files indicated that no obvious problems occurred and that the CPU time spent in event location was increased by 12%. A 40 minute segment was investigated before and after these new travel-time handling facilities were incorporated. All events built with identical association sets in the two tests produced locations within 1 km. of each other.

2.2.3.2 ARS

Extensive interactive testing of **ARS** was conducted and no failures that could be associated with the event location or, more general, travel-time handling facilities were noted. **ARS**, in particular, exploits a far larger number of travel-time handling routines than most IDC applications. For this reason it is a good candidate for robustness testing. Results offered in section 2.2.1.1, *Validation of Default Configuration*, were evaluated as a step in this overall **ARS** testing. It should be noted that the four events located within ARS during release testing were also re-run, as a post-process, using EvLoc. The resulting event locations were exactly the same, to numerical precision, validating, not only the correct integration of the new travel-time handling facilities between these two very different applications, but also verified a consistent configuration as well.

2.2.3.3 EvLoc

Most unit testing provided within this document was done using **EvLoc**. Robustness testing is particularly well suited to **EvLoc** since it is often employed as a batch event location process. Of the many dozens of runs executed (each requiring hundreds of event locations), none ever crashed or displayed any unexpected behavior.

We conducted stress tests as part of our testing in section 2.2.2.1, *Idealized Synthetic Data*, whereby all of our SSSC grids were oversampled with 93 latitude nodes by 67 longitude nodes (*i.e.*, every station/phase includes an SSSC grid containing over 6200 SSSC samples). Furthermore, 98% of all time defining associations exist within a valid SSSC region (refer back to Table 1 for details) thereby exercising access to these regions on almost every travel-time instance. Given these conditions, a 67% increase in computational performance was acknowledged in running the location process for VM_{mimic_fenno} relative to VM_{fenno_only}. This is quite significant, but not likely to be realized in an operational setting for a very long time, if ever. It should be noted that overall EvLoc run time was only increases about 10%. Regardless, while this is not likely to affect an application like **ARS**, it could have a significant impact on the **GA** sub-system. Such comprehensive run-time testing is recommended when these facilities are to be fully implemented in **GAassoc** and **GAconflict**.

2.2.3.4 DFX

DFX was run through a sequence of tests as part of its release testing. No crashes or unusual travel-time behavior was noted. This testing is notable in that, unlike the above applications, **DFX** only requires some approximate travel-time handling routines in order to determine an appropriate processing segment. These routines are also employed by **XfkDisplay** and **WaveExpert**.

2.3 Testbed Testing

These new travel-time handling facilities were inspected as installed on the IDC testbed to hopefully minimize risks associated with the overall integration process. This was accomplished by visually inspecting configuration and log files on the IDC testbed for a number of applications. The inspection of par, scheme and data files validated correct installation procedures were followed as prescribed in the PIDC v. 4.0.0 release notes. Inspection of application log files did not reveal evidence of any clear travel-time related error messages indicative of serious problems.

Appendix A Travel-Time Handling Facilities

A.1 OVERVIEW

The new travel-time handling facilities detailed within this section are primarily controlled by a single high-level velocity model specification file (VMSF). This file is the user's main interface to the four major station/phase-dependent travel-time contributions. This file also functions as the location where the default travel-time tables, as well as the long-period, extremely path-dependent phases, LR and LQ, are specified. The user may specify any or all of the following additive travel-time contributions:

- One-dimensional travel-time tables
- Bulk static station corrections for travel-time
- Sedimentary velocity for station/phase pairs (for better elevation corrections)
- Source-specific station correction (SSSC) travel-time terms

If no information is provided for a given station and phase, then the resultant travel time will only be a function of the user-defined default travel-time tables. Two additional default modes may be employed to apply any of these contributions on a station-dependent only basis if the last leg of a given phase is either a P or an S (the specifics are detailed in Appendix B). The long-period phases, LR and LQ, can now be ray traced on a path-dependent basis where group velocities can be specified on a latitude/longitude grid (Appendix D). The new format for specifying ellipticity corrections is detailed in Appendix C.

In addition, the user may now specify phase-dependent travel-time modelling error knowledge within the travel-time tables themselves as either a single value, a set of distance-dependent values, or a set of distance/depth-dependent values. These modelling errors can be "added" to the measurement error which is stored in the ***deltim*** attribute of the ***arrival*** table. In the past, the ***deltim*** field has been used to estimate an average travel-time modelling plus measurement error. This was probably inadequate.

A.1.1 Motivation for Incremental Approach

The principal goal of developing a methodology for applying source-dependent knowledge via separate correction terms is to provide an incremental approach whereby individual contributions can be measured against one another. Travel-time calculations will be a combine local (station-dependent) velocity model, an ellipticity correction, an elevation correction (with an independent sedimentary velocity defined), a bulk static station correction term, and a source-dependent station correction. By permitting multiple travel-time models to be specified for a given station/phase based on the local velocity structure, one would suspect that any additional travel-time corrections would be minimized. This is particularly important for regional branches, since it is quite likely that a local velocity model represents the largest adjustment to the travel time. This can be taken one step further if a bulk static station correction term is added as well. In very many circumstances, the inclusion of a local one-dimensional model and a single-valued bulk station correction term will be sufficient to describe the travel time. In other cases, more detail will be necessary. In the context of this model, the SSSC terms simply provide the final adjustment information.

A.2 STRUCTURAL ELEMENTS OF APPENDICES

The structural elements of the new travel-time handling facilities are broken down into appendices as defined in Table 9. Note that Appendix F is only included here for general reference. It is not officially sanctioned as part of this implementation effort.

Table 9: Contents of Appendices

APPENDIX	CONTENTS
A	This section. Provides high-level outline of travel-time handling facilities
B	Defines details of velocity model specification file
C	Phase-dependent travel-time, modelling error and ellipticity correction facilities
D	Long-period travel-time facilities to ray trace path-dependent LR and LQ phases.
E	Describes SSSC file structure, contents, and model parameterization(s)
F	A description of conventions used throughout this document
G	Sample velocity model specification file (VMSF)
H	Sample travel-time and modelling error table file
I	Sample long-period travel-time grid and group velocity table files
J	Glossary of terms and acronyms used in this document
K	References

Appendix B Velocity Model Specification File

The velocity model specification file (VMSF) is the central control component for all of the new travel-time handling facilities. We will not only specify all the travel-time table models here, but also detail all station/phase-dependent knowledge within this file. Collecting all this information into just one file allows the user to localize this knowledge thereby permitting a single high-level environment variable to describe, not only an entire set of travel-time tables, but also all source-dependent knowledge. A sample VMSF is provided in Appendix G. Table 10, provided at the end of this appendix, outlines the formatting details of the velocity model specification file.

B.1 SPECIFYING DEFAULT INFORMATION

There are three pieces of default information that can be specified within the velocity model specification file: the location of the default travel-time tables; the location of the default long-period path-dependent phases, LR and LQ; and an optional default station-dependent description defining specific attributes to be assigned to the last P and S-wave legs for a broad spectrum of phases. For this latter case, for example, one may wish to assign the same bulk static station correction and sedimentary velocity for a given station for all teleseismic phases where the last leg is a P-wave type. A default formatting mode is provided for this purpose.

The first uncommented line of the VMSF defines the directory pathway containing the path-dependent long-period LR and LQ travel-time tables (Appendix D). These special tables are assigned group velocities prescribed within pre-defined latitude/longitude regions and ray traced across the Earth's surface using this path-dependent knowledge. The second uncommented line of the velocity model specification file specifies the name and directory pathway location for the default set of travel-time tables. That is, if no station/phase-dependent representation is prescribed by the user, then only these curves will be employed. If the directory pathway does not exist, then the reading routines will return with an imperative error message. Please note that the terms "velocity model" and 'travel-time tables" are used synonymously, since the travel-time tables are simply the physical parameterization used to represent the velocity models. It's worth repeating that this default set of curves must be specified on the second uncommented line of this file. Additional velocity models may be specified on subsequent lines until a blank line is encountered indicating all models have been read and station/phase-dependent knowledge follows. A comment line (represented by a '#' sign in the first column of any line) may be employed throughout this file for the sake of clarity. Such comment lines are highly encouraged.

A special default mode exists for prescribing default station-dependent knowledge for any phase whose last leg may end as either a P- or S-wave. If a given station/phase specification has been made (as defined in the next section) the phase may be described as *P or *S indicating that any phase for that station whose last leg (branch) ends as a P or S will have the knowledge specified on that line applied to that phase type. For example, if a given station is assigned a phase type of *P, then the velocity model, sedimentary velocity and bulk station correction defined by this default mode will be employed for such phases as: P, PKPdf, PKKPab, Pn, P7KP, etc.

B.2 STATION/PHASE-DEPENDENT KNOWLEDGE DESCRIPTION

In this section we describe how station/phase-dependent knowledge is incorporated into our travel-time handling facilities This information is specified immediately after all the velocity model descriptions have been defined. Each line of this station/phase-dependent information contains, in order, the station name, phase type, velocity model name (set of travel-time tables), sedimentary velocity and bulk static station correction. The previous section outlined the behavior if the phase type was defined by the default modes, *P and/or *S. Here we will assume it is a specific phase type.

It is important to realize that all corrections made as part of the travel-time handling facilities described here are additive and directional. They are directional in the sense that there is a fixed precedence: ellipticity corrections are applied before bulk static station corrections, and both these corrections before SSSC terms. This is important to preserve data consistency. The algorithmic order for applying corrections is ellipticity, elevation, static station correction, and finally, SSSC. Therefore, if SSSC terms are to be applied, then the other three corrections must exist. If test-site or SRST corrections are to be applied, then only ellipticity and elevation corrections are added. The details of the test-site and SRST corrections are not presented here.

Before these corrections are attempted, the travel time is interpolated from the set of travel-time tables based on a one-dimensional Earth model as specified by the velocity model name. The velocity model name must be identical to one of the velocity models previously attached to a directory location. It not, a fatal error will be returned along with an informative message.

The ability to specify station-dependent sedimentary velocity allows elevation corrections to be more representative of the local geological structure. A default sedimentary velocity of 5.8 km/s and 3.35 km/s will be used if no specific information is detailed for a given station and phase. Finally, if specified, a bulk static station correction and SSSC is also added to the travel time. The bulk static station correction (seconds) is added to the correction for elevation.

If SSSC information is desired for a given station and phase, then this information must be duplicated within the associated SSSC file itself, even if the default velocity model and sedimentary velocity are to be used without bulk static station corrections. SSSC files will be searched for every station/phase-dependent entry included in this file. By prescribing this structure, an SSSC file can simply be omitted if it is not desired. It is imperative that an entry exists within the VMSF for every SSSC file to ensure data consistency. This is discussed in detail in Appendix E. Suffice it to say here, that since all corrections are additive and directional, if one wishes to apply later corrections (such as SSSC terms), then earlier corrections must be defined. Multiple SSSC source regions may exist for a single station/phase pair. Details about how this structure helps simplify multiple SSSC source regions is described in Appendix E.

B.3 SOURCE CODE AND STRUCTURE DEFINITIONS

All source code necessary to read information contained with the velocity model specification file is located in library, *libsrc/libloc/*. This code is written entirely in the ANSI-compliant C programming language following the style of [Kernighan and Ritchie 1988]. All reading of the velocity model specification information is done within file, *libsrc/libloc/read_tt_tables.c*. Application of this knowledge is accommodated in function, `total_travel_time()`, located in file, *libsrc/libloc/trv_time_specific.c*.

The functions are all linked together using the model description, list of phases, and station/phase/model structures, model_descrip (type, `Model_Descrip`), list_of_phases (type, `List_of_Phases`), and sta_phase_model (type, `Sta_Phase_Model`), respectively, as defined in the include file, *libsrc/libloc/tt_table.h*. These structures are defined as:

```
typedef struct  model_descrip    Model_Descrip;
struct model_descrip {
         char              vmodel[16];
         char              *dir_pathway;
         List_of_Phases    *list_of_phases;
         Bool              sssc_dir_exists;
};

typedef struct  list_of_phases    List_of_Phases;
struct list_of_phases {
         char              phase[9];
         int               phase_index;
         List_of_Phases    *next;
};

typedef struct  sta_phase_model    Sta_Phase_Model;
struct sta_phase_model {
         char              sta[7];
         char              phase[9];
         int               phase_index;
         int               vel_index;
         float             sed_vel;
         float             bulk_sta_corr;
         Sssc              *sssc;
```

```
};
```

Only the model_descrip structure is populated as the velocity name and directory pathway information is read. At this point, its *list_of_phases* variable is NULL and *sssc_dir_exists* attribute is FALSE. As the station/phase-dependent knowledge is read, one line at-a-time, the sta_phase_model structure is populated. The *vmodel* field is associated with its model_descrip structure, and the *phase* is added to the related *list_of_phases* link list, if it does not already exist there. The *vel_index* field links structure, sta_phase_model, with structure, model_descrip.

To expedite access to this information, two additional structures, sta_pt (type, `Sta_Pt`) and phase_ptr (type, `Phz_Pt`), are employed. When the station and travel-time information is initially encountered, these structures are established, thereby providing immediate access to the sta_phase_model structure via the *spm_index* field. Therefore, once the initial station/ phase information is indexed, all the information defined in this section is instantly available, even the pointer to the `SSSC` file. This linkage is extremely efficient. These two new structures are simply defined as:

```
typedef struct  phase_ptr    Phz_Pt;
struct phase_ptr {
         char      phase[9];
         int       spm_index;
         Phz_pt    *next;
};

typedef struct  sta_pt    Sta_Pt;
struct sta_pt {
         Phz_pt    *phase_ptr;
};
```

B.4 FORMAT DESCRIPTION

This section outlines the general format description for the contents of velocity model specification file. The line number field assigned in Table 10 excludes comment lines. (**vmodels** is the number of velocity models defined, including the default travel-time table model).

Table 10: Velocity Model Specification File Description

LINE NUMBER	FORMAT STATEMENT	DESCRIPTION
1	%s	Relative directory pathway for path-dependent long-period info
2	%15s%s	Velocity model name and relative directory pathway containing default travel-time tables. Velocity model name cannot be longer than 15 characters
3 thru **vmodels** + 1	%15s%s	Velocity model name and relative directory pathway containing travel-time tables assigned to given model
vmodels + 2	\n	A blank line indicates all velocity models (vmodels) have been defined. Required to separate station/phase-dependent knowledge that follows.
vmodels + 3 thru EOF	%s%s%s%lf%lf	Contains, in order, station name, phase type, velocity model, sedimentary velocity (km/s), and bulk station correction (s)

Appendix C Travel-Time Tables

The definition of the travel-time tables themselves has not changed, but phase-dependent travel-time modelling errors can now be optionally appended to these traditional files. These additional modelling error terms can be single-valued, distance-dependent or distance/depth-dependent based on the user's *a priori* knowledge. Furthermore, newly incorporated two-dimensional ellipticity correction tables are directly attached to the travel-time table structure.

C.1 TRAVEL-TIME TABLES

While the travel-time tables have existed in their present form for many years, they have never been formerly defined. Now that these table can be scattered in many different directories controlled only by the VMSF, we can easily include or exclude any specific table with great ease (unintentionally if we're not careful). Furthermore, the fact that travel-time modelling errors have been added (see below) demands a clear formatting description.

C.1.1 Parameterization

Travel-time tables contain travel-times (seconds) for a given phase type discretized into two-dimensional spatial coordinates of distance (arc degrees) and depth below sea level (kilometers). Sampling can be of non-uniform spacing, but since bi-cubic splines are used for interpolation purposes, must be continuous across the grid. Sample points for which no clear geometric ray path exists are indicated with a entry of -1.0. Bi-cubic spline interpolation is chosen to determined the travel time and associated derivatives within the grid. The global nature of bi-cubic spline interpolant function, along with the property that all its second derivatives be continuous everywhere, assists in smoothing out curves that are, in reality, separated by abrupt changes in slowness. This is particularly accentuated when new branches are encountered with distance and depth. The advantage is also a bit of a drawback in that resultant derivatives are less localized. But the benefit of eliminating oscillating solutions from iteration-to-iteration in the non-linear least-squares inversion outweighs the loss of localized information, especially in cases when this localization is undesirable.

C.1.2 Travel-Time Modelling Errors

Provided a phase-dependent travel-time modelling error does exist, then the "total" *a priori* error is composed of the measurement variance plus the travel-time modelling variance (*i.e.*, the RMS of measurement and modelling errors). To preserve backward-compatibility, if no modelling errors are specified, then only the measurement error, as determined from the ***arrival*** attribute, ***deltim***, comprises the "total" *a priori* error. In this case, users should make sure that the phase-dependent travel-time modelling error (**dist_var_wgt**) is set FALSE, since this control parameter dictates whether these modelling errors are to be applied. Optimally, ***deltim*** should be adjusted to only represent the measurement error in applications such as **DFX** and **SigPro**. When this feature is implemented, then the distance variance weighting should always be set TRUE so that if phase-dependent modelling errors do exist, they get applied. The inverse "total" *a priori* error is multiplied as a weight to its corresponding row in the event location sensitivity matrix. Therefore, the larger the error, the smaller the weight assigned to that datum (*i.e.*, station/phase).

The phase-dependent modelling errors can be specified as single-valued, distance-dependent or distance/depth-dependent terms. Distance-dependent and distance/depth-dependent modeling errors are linearly (or bi-linearly) interpolated from the discrete values stored in the travel-time tables.

C.1.3 File Management

Each travel-time table, including its modelling errors, is limited to one phase type and is stored in an ASCII data file. The directory pathway containing this file is obtained from the `VMSF`. If not specified, these tables will not be read. An ASCII travel-time table file follows the file naming convention, VMODEL.PHASE, where VMODEL is identical to the velocity model name specified in the velocity model specification file and PHASE is the phase type of interest. This phase type will only be read if requested by the calling routine. For example, in ARS(1), the list of phases to be read is determined by the scheme variable, `list-of-phases`. Hence, if the velocity model defined in the `VMSF` was 'ak135' and the current phase type in the list of phases was 'PKPdf', then the file to be searched for would be, 'ak135.PKPdf', under the directory pathway linked to the velocity model name.

C.1.4 Source Code and Structure Definitions

All source code necessary to read travel-time and modelling error information is located in library, *libsrc/libloc/*. All reading of this information is done within file, *libsrc/libloc/read_tt_tables.c*. Travel times are interpolated from their two-dimensional table via function, `trv_time_w_ellip_elev()`. Travel-time modelling errors are obtained using function, `get_model_error()`. Both of these functions are located in the file, *libsrc/libloc/trv_time_specific.c*.

The two corresponding structures, tt_table (type, `TT_Table`) and model_error (type, `Model_Error`) are defined in include file, *libsrc/libloc/tt_table.h*. Note that the associated model_error and ellipticity correction (ec_table; described in the next section) structures are physically attached to the tt_table structure for rapid access. If no modelling error or ellipticity correction tables are available, then the pointers to these two structures will be set to NULL. If only a single valued modelling error exists, then only the *bulk_var* variable needs to be populated in the model_error structure. Likewise, if only distance-dependent errors exist, then *num_dists* and *dist_samples* is filled. If distance/depth-dependent modelling errors are available, then the entire model_error structure will be populated. These two structures are defined on the following page as:

```
typedef struct  tt_table    TT_Table;
struct tt_table {
         char             phase[9];
         char             vmodel[16];
         int              last_leg;
         int              num_dists;
         int              num_depths;
         float            in_hole_dist[2];
         float            *dist_samples;
         float            *depth_samples;
         float            **trv_time;
         EC_Table         *ec_table;
         Model_Error      *model_error;
};

typedef struct  model_error    Model_Error;
         float            bulk_var;
         int              num_dists;
         int              num_depths;
         float            *dist_samples;
         float            *depth_samples;
         float            *dist_var;
         float            **dist_depth_var;
};
```

C.1.5 Format Description

This section outlines the general format description for contents of a travel-time table file. Table 11 provides the file description for the travel-time table and, if available, the first line of the travel-time modelling error information. Since these errors may be defined as a single value, as distance-dependent values, or distance/depth-dependent values, Tables 12, 13 and 14, represent these three respective cases. In Table 11, **I** is the current depth sample from 0 to **nz**-1. **J** represents the first line of travel-time modelling information in all four tables. Intermediate distance-dependent modeling errors are linearly interpolated from the discrete distance sampled modeling errors defined in Table 13. Intermediate distance/depth-dependent modeling errors are bi-linearly interpolated from the discrete distance/depth sampled modeling errors defined in Table 14.

Table 11: Travel-Time Table File Description

LINE NUMBER	FORMAT STATEMENT	DESCRIPTION
1	\n	Comment line (typically identifies velocity model and phase)
2	%d%*[^\n]	Number of depth samples (**nz**) in given travel-time table
3	**nz***%f	Depth samples (in km below sea level)
4	%d%*[^\n]	Number of distance samples (**nx**) in given travel-time table
5	**nx***%f	Distance samples (in arc degrees = 111.195 km.)
6 + **I*nx**	\n	Comment line typically indicating current depth sample.
(7 + **I*nx**) thru (5 + (**I**+1)***nx**)	%f	Travel-time (seconds)
6 + **nz*nx**	\n	Comment line or EOF if no modelling errors available
J = 7 + **nz*nx**	%d%d	Number of distance (**mx**), depth samples (**mz**) for travel-time modelling errors

Table 12: Travel-Time Modelling Errors File Description (Single Value Case)

LINE NUMBER	FORMAT STATEMENT	DESCRIPTION
J + 1	\n	A comment line. By definition, **mx** = 1 and **mz** = 1 here
J + 2	%f	Travel-time modelling error (seconds)

Table 13: Travel-Time Modelling Errors File Description (Distance-Dependent Case)

LINE NUMBER	FORMAT STATEMENT	DESCRIPTION
J + 1	**mx***%f	Modelling error distance samples (arc degrees). By definition, **mz** = 1 here
J + 2	\n	Comment line
J + 3 thru **J** + 2 + **mx**	%f	Travel-time modelling error (seconds)

Table 14: Travel-Time Modelling Errors File Description (Distance/Depth-Dependent Case)

LINE NUMBER	FORMAT STATEMENT	DESCRIPTION
J + 1	**mx***%f	Modelling error distance samples (arc degrees)
J + 2	**mz***%f	Modelling error depth samples (km below sea level)
J + 3	\n	Comment line
J + 4 thru **J** + 3 + **mx**	%f	Travel-time modelling error (seconds). Repeat last two rows outlined here **mz** times.

C.2 NEW ELLIPTICITY CORRECTION TABLES

Besides the ability to directly include phase-dependent travel-time modelling errors in these curves, we now also directly attach the newly-defined two-dimensional ellipticity correction tables of Kennett (1995; unpublished) for many more phases then was previously practical. For example, the new AK135 model of Kennett *et. al* (1996) provides these detailed ellipticity correction tables for over 50 phases, including depth phases which were previously not available. The lack of ellipticity corrections for depth phases can result in a significant depth bias if the main phases do have these corrections applied.

C.2.1 Derivation of Ellipticity Corrections

Ellipticity corrections are applied via three coefficient terms, τ_0, τ_1, and τ_2, stored within the ellipticity correction tables at pre-specified distance/depth sample points following the formulation of Dziewonski and Gilbert (1976). Given the co-latitude of the source as θ, and the distance and azimuth from the epicenter to the station as Δ and ζ, respectively, the ellipticity correction is given by the formula:

$$\delta t = \frac{1}{4}(1 + 3\cos 2\theta)\tau_0(\Delta) + \frac{\sqrt{3}}{2}\sin 2\theta \cos\zeta\tau_1(\Delta) + \frac{\sqrt{3}}{2}\sin^2\theta\cos 2\zeta\tau_2(\Delta)$$

C.2.2 File Management

Each ellipticity correction table is limited to one phase type and is stored in an ASCII data file. The directory pathway containing this file is obtained from the VMSF with the string, ‘Ellip/’ attached to it. If not specified, these tables will not be read. An ASCII input ellipticity correction table file follows the file naming convention, ec_VMODEL.PHASE, where VMODEL is identical to the velocity model name specified in the VMSF and PHASE is the phase type of interest. This phase type will only be read if its associated travel-time table has been read. For example, in ARS(1), assuming the travel-time table for phase, PKPdf, had been read, the velocity model defined in the `VMSF` was 'ak135', and the current phase type in the list of phases was 'PKPdf', then the ellipticity correction file to be searched for would be, 'Ellip/ec_ak135.PKPdf', under the directory pathway linked to the velocity model name.

C.2.3 Source Code and Structure Definitions

The ellipticity correction tables have been made to look much like a traditional travel-time table, except that there are three entries (coefficients) at every distance/depth point instead of just one. These tables are directly attached to the corresponding tt_table structure, defined in the previous section, via a pointer. If the pointer is NULL, then no ellipticity corrections are available. Therefore, once the travel-time table is established the associated ellipticity correction information is immediately available.

All source code necessary to read ellipticity correction information is located in library, *libsrc/libloc/*. All reading of this information is done within file, *libsrc/libloc/read_tt_tables.c*. Corrections are read from their two-dimensional tables via function, `read_tt_tables`. They are employed using function, `get_ec_from_table()` from file, *libsrc/libloc/ellipticity_corr.c*.

The structure, ec_table (type, `EC_Table`) is defined in include file, *libsrc/libloc/tt_table.h*, as:

```
typedef struct  ec_table    EC_Table;
         int               num_dists;
         int               num_depths;
         float             *dist_samples;
         float             *depth_samples;
         float             **t0;
         float             **t1;
         float             **t2;
};
```

C.2.4 Format Description

This section outlines the general format description for the contents of ellipticity correction table file. Table 15 provides the file description for the ellipticity correction table. **I** is the current depth sample from 0 to **nz**-1.

Table 15: Ellipticity Correction Table File Description

LINE NUMBER	FORMAT STATEMENT	DESCRIPTION
1	\n	Comment line (typically identifies velocity model and phase)
2	%d%*[^\n]	Number of depth samples (**nz**) in ellipticity correction table
3	**nz***%f	Depth samples (in km below sea level)
4	%d%*[^\n]	Number of distance samples (**nx**) in ellipticity correction table
5	**nx***%f	Distance samples (in arc degrees = 111.195 km.)
6 + **I***nx**	\n	Comment line typically indicating current depth sample.
(7 + **I***nx**) thru (5 + (**I**+1)***nx**)	%f%f%f	Ellipticity correction τ coefficients, τ_0, τ_1, and τ_2

Appendix D Path-Dependent LP Data

Path-dependent long-period travel time calculations for LR and LQ phases is new to this release of the our travel-time handling facilities. LR and LQ phases are ray traced on a path-dependent basis across the Earth's surface where group velocities can be specified on a latitude/longitude grid. These travel times are path-dependent in that they are ray traced within each cell along a great circle surface path as specified by the individual cells group velocities. All such segments are added together to obtain the total path-corrected travel-time.

D.1 ALGORITHM

Long-period phases are ray traced over a grid representing group velocity versus period measurements mapped onto the Earth's surface along a great circle path as prescribed by the following algorithm:

1. Given a geographic event and station location and desired period (seconds), always start at the westernmost so that we always trace from west to east. This will assure that the ray never exits from the west side of a longitude grid boundary.
2. We begin by ray tracing from the current starting point to both northeastern and southeastern corners of the grid cell. This establishes our reference points.
3. If the current azimuth lies between two computed corners, then we know ray exits from eastern grid boundary. We then calculate the distance and new latitude/longitude position along the east longitude grid boundary where it intersects with the latitude. If the current azimuth lies between the southeastern corner and 180 degrees (due south), then the exit point lies along the southern grid boundary. Otherwise, the exit point falls along northern boundary. In these last two cases, we calculate the distance and new position to the southern or northern latitude grid boundary based on its longitude intersection.
4. If the distance to the exit boundary is greater than the current remaining distance, then the final grid cell has been found.
5. Compute the travel-time spent within this segment of the ray path based on the velocity for the given period. Add this segment to the previous segments computed thus far. If the final grid cell has been found, go to Step 7.
6. Calculate a new "current" azimuth and distance from our new starting point to the final easternmost point (which may be exiting from the north, south or east boundary). This will provide the distance for testing Step 6 during the next iteration. Go to Step 2.

7. Ray path completely traced. Check that the sum of distances for each individual grid cell is nearly equal (within 0.1%) to the original great circle distance computed between the station and the event. This check is just a verification that the ray was traced properly. If successful, return travel time; else return with an error code.

D.2 Long-Period Group Velocity vs. Period Tables

The table information necessary to compute LR and LQ travel times are constructed from two separate files for each phase type. The first “grid” file stores group velocity indexes discretized by latitude and longitude. Grid spacing must be the same in both directions. The second “velocity” file maps these indexes to their respective group velocity versus period measurements, which are stored here.

D.3 File Management

The filesystem structure is rather rigid in that the file names are fixed, although the long-period directory is specified by the user in the velocity model specification file as outlined in Appendix B. The LR and LQ grid files must be named, ‘LP_grid.LR’ and ‘LP_grid.LQ’, respectively, within the prescribed directory. The corresponding velocity files must be named, ‘LP_vel.LR’ and LP_vel.LQ’, respectively. This will likely change in the future, but for now its rigid. This should not be a problem, since I don’t imagine anyone will have any reason to change these for some time.

D.4 Source Code and Structure Definitions

All source code necessary to read and ray trace path-dependent long-period phase information is located in library, *libsrc/libLP/*. All reading of this information is done in function, `read_LP_info()`, located within file, *libsrc/libLP/LP_trace_ray.c*. Path-dependent travel times are ray traced in function, `LP_trace_ray()`, also located within file, *libsrc/libLP/LP_trace_ray.c*. Furthermore, grid indexes and velocities are extracted by calls to functions, `get_LP_grid()` and `get_LP_velocity()`, respectively, again located within file, *libsrc/libLP/LP_trace_ray.c*.

The structure, lp_data (type, `LP_Data`) is defined in include file, *libsrc/libLP/lp_data.h*. This structures is defined as:

```
typedef struct  lp_data    LP_Data;
struct lp_data {
        int                num_lat_grids;
        int                num_lon_grids;
        double             latlon_spacing;
        short              **grid_indice;
        int                num_periods;
        int                num_indexes;
        double             *period_samples;
        double             **velocity;
};
```

D.5 FORMAT DESCRIPTION

This section outlines the general format description for the contents of the long-period grid and velocity files. Table 16 provides the file description for the long-period grid index table and Table 17 defines the contents of the long-period group velocity versus period table.

Table 16: Grid Index Table File Description

LINE NUMBER	FORMAT STATEMENT	DESCRIPTION
1	\n	Comment line
2	%d%d%lf	Number of latitude (**nlat**) and longitude (**nlon**) samples and spacing between latitude and longitude grids (degrees)
3	\n	Blank line
4	\n	Comment line typically indicating current co-latitude range
5	**nlon***%d	Grid index indicating which set of group velocities to use from long-period velocity file
6	\n	Blank line
Repeat	Repeat	Repeat Line Numbers 4 thru 6 **nlat**-1 more times

Table 17: Group Velocity vs. Period Table File Description

LINE NUMBER	FORMAT STATEMENT	DESCRIPTION
1	\n	Comment line (typically indicating phase)
2	%d	Number of index samples (**nsamp**)
3	%d	Number of period samples (**nper**)
4	**nper***%lf	Period samples (seconds)
5	\n	Comment line typically indicating current velocity index
6 thru 6 + **nper** - 1	**nper***%lf	Group velocity for given grid index and period sample
Repeat	Repeat	Repeat Line Numbers 5 thru (6 + **nper** - 1) **nsamp** more times

Appendix E SSSC Implementation Details

This appendix describes the current implementation for addressing source-dependent travel-time knowledge via the source-specific station correction (SSSC) methodology. SSSC terms are applied in an attempt to mimic the Earth's three-dimensional velocity field by employing a flexible and efficient parameterization relative to a simple one-dimensional Earth model.

E.1 MODEL PARAMETERIZATION

Two model parameterizations are described here, although only the first is implemented. The second parameterization is documented here because of its similarity to the rectangular grid structure currently used. This secondary circular grid parameterization is just offered for consideration.

E.1.1 Rectangular Grid (Implemented)

As currently implemented, source-specific station correction (SSSC) tables are specified by a parameterization along latitudinal and longitudinal nodes sampled as prescribed by the user (Figure 1; top). A travel-time table is required for each phase with all corrections being measured relative to this table. The requirement that each source region smoothly transition into the background one-dimensional model at its boundaries lends itself nicely to bi-cubic spline interpolation. Bi-cubic splines ensure that second derivatives will be continuous everywhere. Therefore, defining a source region with a 0.0 eqipotential boundary condition along its entire outer edge will guarantee complete continuity across the boundary. As a general policy, all boundary nodes (indicated by open circles in Figure 1) should have 0.0 corrections. The gray area in this figure represents the tapered zone between the actual corrections and the boundary. You should make this wide enough so the spline isn't forced into overshooting, especially if your corrections are large along the edges of the white area (*i.e.*, your actual corrections). The station can be located either inside or outside of your source region. The file format is described in the next section. Since rectangular grids are employed, determining which grid region, if any, the event is located inside, is a very fast operation. User's cannot specify an SSSC file for path-dependent long-period phases, such as, LR and LQ.

E.1.2 Circular Grid (Not Implemented)

Defining our corrections to strictly fall on a rectangular latitude/longitude grid limits its dimensions to regional distances. An alternative simple circular parameterization is offered for consideration (Figure 1; bottom). This parameterization is chosen because the current SSSC grid structure could be largely preserved. Instead of representing the data in terms of latitude and longitude, grid nodes are defined by distance (deg.) and azimuth (deg.). from the center, (by definition, the origin). Boundary conditions still require 0.0 corrections at the boundaries and a taper is employed, but it is now a function of distance. Of course, grid points are much further apart towards the outer edges. A more complex description of the nodal locations would complicate the interpolation scheme. But this parameterization would be more inclusive, making specification of a source region near the poles easier. Further, such a parameterization could more naturally transition into the tectonic regionalization approach of Sambridge *et. al* (1995). It too, though, is primarily limited to regional distances. For this reason, it is not clear whether the introduction of this parameterization is worth the development effort that would be required.

E.2 PRESERVING DATA CONSISTENCY

Paramount to this design was the goal of ensuring data consistency using the VMSF as the central user control interface. To accomplish this goal we must make sure that changes to station/phase-dependent information within this file do not negate knowledge implicit in previously-determined SSSC files. For example, the user determines that a new bulk static station correction should be employed for station X. Since our corrections are additive and directional we must make sure previous travel-time corrections in our SSSC file for station X are properly adjusted to reflect this change. This is accomplished by duplicating critical knowledge (namely, the velocity model, bulk static station correction, sedimentary velocity and source region) in the VMSF within the SSSC file itself. This knowledge is actually stored on the first uncommented line of the SSSC file. If a difference is noted, then the SSSC values are statically adjusted and the user is notified that a new SSSC file has been created for them and preserved in the system /tmp area. Periodically, these new files should be installed in the operational area, although the only real harm results from repeated warning messages when these facilities are used. The new files will have the same name as the original with the string, '_new', appended to them.

E.3 SOURCE-DEPENDENT MODELLING ERRORS

We should expect that if SSSC information is employed in our travel-time determination that the phase-dependent modelling errors we assumed within the travel-time tables are now too large. Therefore, a modelling error correction factor (**me_factor**) is defined as the last argument on the first uncommented line of the SSSC file. It will be multiplied to the phase-dependent modelling error obtained from the travel-time tables to reflect an 'average' reduction in the overall geographic SSSC region. This factor should generally vary from 0.0 to 1.0, but can be larger. Ideally, we would have a separate modelling error at every grid point in the SSSC region, but this would require much more memory and knowledge about the distribution of these errors. This knowledge is not likely to be immediately available and the memory is costly. It would not be difficult to include such information in the grid; I just see no overriding demand for it at this time. It will be taxing enough to come up with a single factor.

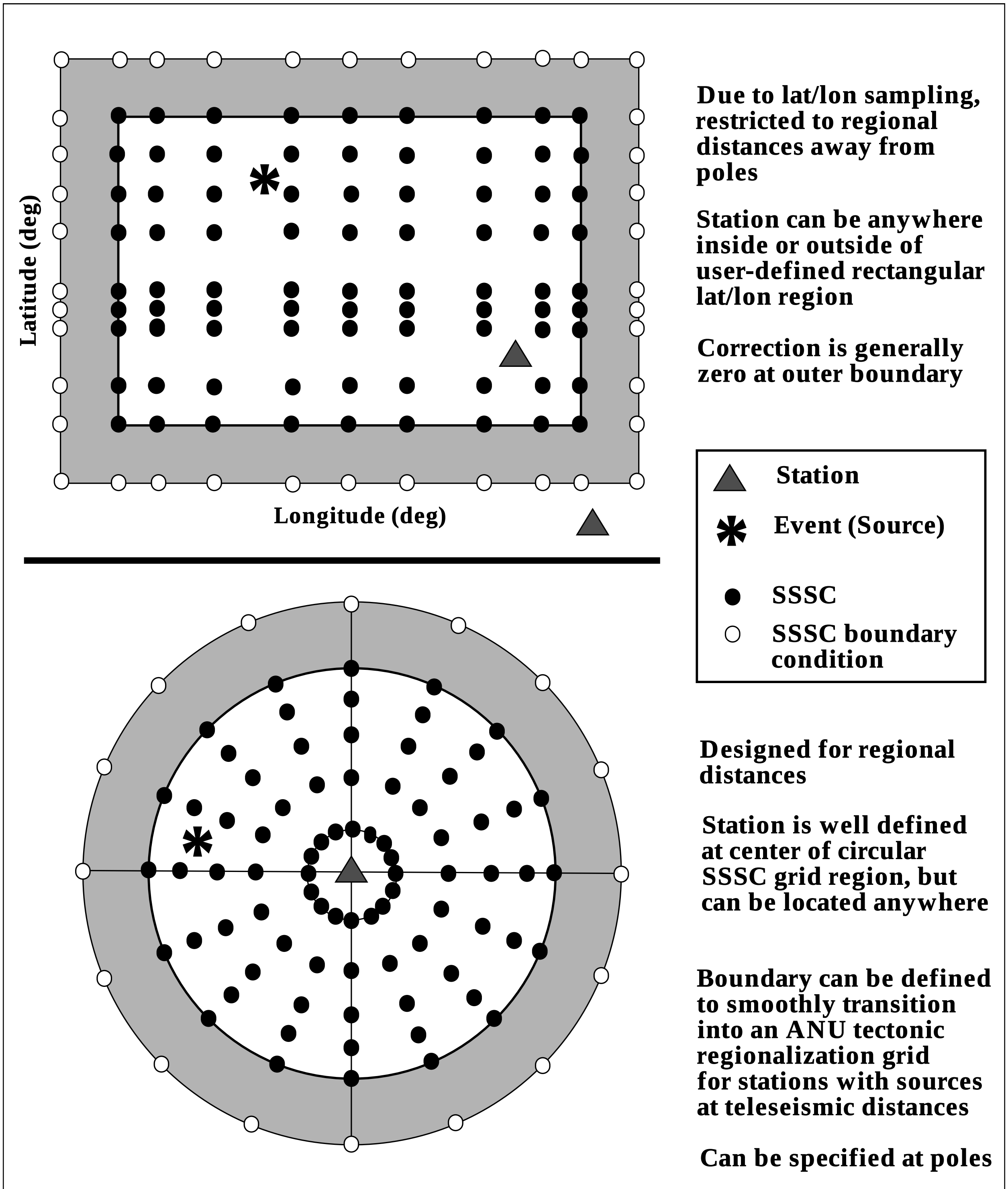

Figure 1: Two SSSC model parameterizations. Top: Rectangular grid model specified by latitude and longitude nodes with origin in NW corner. Bottom: Circular grid model defined by distance and azimuth from center (origin). Solid circles represent actual SSSC's while open circles indicate boundary condition. Grey areas is where taper is applied.

E.4 SOURCE CODE AND STRUCTURE DEFINITIONS

All source code necessary to read SSSC files and apply their corrections are located in library, *libsrc/libloc/*. The SSSC code is written entirely in the ANSI-compliant C programming language following the style of [Kernighan and Ritchie 1988]. In fact, all manipulation of these corrections is done in one file, *libsrc/libloc/sssc.c*. This includes the reading of the SSSC files themselves via function, `read_sssc()`, as well as the actual application of these corrections in function, `apply_sssc()`.

The functions are all linked together using the SSSC structure, sssc (type, `Sssc`), as defined in the include file, *sssc.h*. This structure contains all SSSC information and is defined as:

```
typedef struct  sssc    Sssc;
struct sssc {
         char     source_region[16];
         double   me_factor;
         float    origin_nw_lat;
         float    origin_nw_lon;
         float    se_lat_bound;
         float    se_lon_bound;
         float    depth;
         int      nlats;
         int      nlons;
         float    *lat_grid_ref;
         float    *lon_grid_ref;
         float    **sta_cor;
         float    **sssc_2nd_deriv;
         Sssc     *next_src_region;
         Sssc     *finer_detail;
};
```

E.5 INPUT

SSSC information read from two sources: parameters specified by the user, and ASCII data files containing the SSSC's themselves. The sections below describe each source and the type of information they provide.

E.5.1 Parameter Input

Parameter input comes either directly from the command-line or from input parameter (par) files of the form described in [getpar(3)]. This input contains both general run-time parameters and processing parameters. The only pertinent par file argument to this discussion is `sssc_level`. Applications, **EvLoc** and **LocSAT**, read this argument to inform them as to the user's level of desired corrections; if `sssc_level` = 0, no corrections will be applied; if `sssc_level` = 1, then only regional (1st level) corrections will be applied; if `sssc_level` = 2, then local (2nd level; greater detailed) correction will be attempted. If no local level correction exists, then the regional level correction will be applied, if applicable. Refer to [EvLoc(1)] for specific details.

E.5.2 ASCII Data Files

The ASCII input files containing the SSSC data are read by following the file naming convention, DATA_TYPE.STA.PHASE.LEVEL.SOURCE. Currently only travel-time (TT) data can be specified as the DATA_TYPE. PHASE is the phase type, STA is the station name and LEVEL is either, local (local) or reg (regional). SOURCE is the source region name and does not require a specific name. Therefore, regional level travel-time corrections at station, ARA0, for phase, Pn, in Fennoscandia would be contained in a file by the name, 'TT.ARA0.Pn.reg.fenno'. Note the SOURCE can be abbreviated or even given a tectonic, as opposed to regional, name descriptor. A directory explicitly named, SSSC, must reside immediately below the travel-time table directory corresponding to the given station/phase pair if one desires to have these source-dependent corrections applied.

The question arises, why have such a rigid structure? Obviously, we could create extensive file listings with any name we wish. My experience with this degree of flexibility is mixed. While it is indeed convenient to name files anyway a given user desires, I find, all too often, that many files are simply overlooked, or worse yet, outdated. A better alternative is the data consistency checking approach adopted here where the user is explicitly warned about any inconsistencies and even a newly adjusted SSSC file is created for them, again following a rigid naming convention. Furthermore, anyone viewing a directory immediately has available to them the important contents of each and every file. In the above example, the contents of the file are unambiguous.

E.6 FORMAT DESCRIPTION

This section describes the general format description for the contents of SSSC files. Table 18 outlines the details of the SSSC file description.

Table 18: SSSC File Description

LINE NUMBER	FORMAT STATEMENT	DESCRIPTION
1-2	\n\n	Comment lines describing contents of line 3
3	%s%s%s%s%s %lf%lf%lf	Contains, in order, station name, phase type, velocity model, source region, region level, bulk station correction, sedimentary velocity, and modelling error factor
4	%s	Comment line
5	%5d%5d	Number of model nodes in latitude and longitude directions
6	%s	Comment line
7	%10.3f[]	Latitude samples (deg.). North of equator is positive
8	%s	Comment line
9	%10.3f[]	Longitude samples (deg.). East of Greenwich meridian is positive
10	%s	Comment line.
11 thru 11+ #lat nodes - 1	%8.2f[]	SSSC (in seconds). Number of longitude samples per line.

Appendix F Document Conventions

F.1 CONVENTIONS USED IN THIS DOCUMENT

This section describes conventions used throughout this document.

F.1.1 Pathnames

Pathnames to directories and filenames are relative to the top of the software development tree, since other installations are likely to use the same tree structure, but with a different root path. For example, at SAIC, San Diego, the top of the development tree is */prj/shared*. Therefore, the local pathway for say file, */prj/shared/src/EvLoc/src/reset_evloc_controls.c,* would be referenced here as file, *src/EvLoc/src/reset_event_controls.c*. This document uses the common UNIX style convention for upper- and lower-case letters. Often times the moniker, *PRJ_DIR*, is employed to indicate a high-level project directory reference.

F.1.2 Fonts and Formats

The helvetica font is used for acronyms (*e.g.*, IDC and SSSC).

The *italic* font is used for pathname, filenames (*e.g.*, */prj/shared* or *PRJ_DIR*), and libraries.

The simple **bold** font is used for the names of applications (*e.g.*, **ARS**, **DFX** and **EvLoc**).

The ***bold italic*** font is used for the names of database relations and attributes (*e.g.*, ***origin***, ***assoc, lat***, and ***time***).

The `courier` font is used for code samples. This font is also used for variable names, structures, defines, function names, typedefs, etc. (*e.g.*, `read_sssc()`, `Sssc`).

An overline above a word or symbol indicates a mean or average (*e.g.*, $\overline{\textbf{\textit{timeres}}}$).

UNIX manual pages are shown in square brackets, *e.g.*, [EvLoc (1)].

F.1.3 Acronyms

An acronym is defined the first time it appears in the document (*e.g.*, International Data Center (IDC)). Expanded acronyms (*e.g.*, Event Locator (EvLoc)) may also be defined in a like manner. The acronym is used thereafter. Some degree of discretion is taken in referring to some acronyms, for example, more than one SSSC is referred to as SSSC's. While not strictly grammatically correct, it makes writing this document easier. Appendix J lists the acronyms defined in this document.

Appendix G Sample VMSF

This appendix contains a sample velocity model specification file (VMSF). This file is described in Appendix B and was used for unit testing (section 2.2, *Unit Testing*) of velocity model, VM_{mimic_fenno}.

```
#
# The first line of this file defines the directory locations for the path-dependent long-period
# info (namely, LR and LQ phases).

../LP                    "#" "Directory" "location" "of" "path-dependent" "LP" "info" "(LR/LQ)"
#
# The following lines indicate directory locations for specific velocity models defined below.
# The very first line defines the model name and directory location for the default travel-time
# set.  If a velocity model is discovered without a representation here, then the reading routines
# will complain loudly.  Note that the velocity model identifier cannot be more than 15 characters
# long to satisfy the vmodel attribute description contained within the assoc table.  A blank line
# must follow the list of models immediately below.  A '#' can occur in the first column anywhere
# within the list of models for comments.
#
# Velocity Model      Directory location relative to this directory
# -----------------   -----------------------------------------------------
ak135                 ../ak135
# Local velocity models
fenno                 ../local/fenno


# Station/phase specific knowledge.  This includes station/phase-dependent velocity model,
# sedimentary velocity, travel-time modeling error, and a bulk station correction term.
#
# Note that each line of station/phase-dependent knowledge includes, in order, the station name,
# phase type, velocity model, sedimentary velocity (km/sec) for making elevation corrections,
# and a bulk static station correction term (sec.).  The bulk static station correction will be
# added to the overall travel-time relative to the velocity model specified on the same line.
# If no station/phase info resides here, then the default phase-dependent model specified above
# will be used.  Individual line entries only need be space delimited.  Any line with '#' in its first
# position will be ignored throughout this file.  Please be aware that any SSSC files found within
# these travel-time areas will be read as well, regardless of whether or not they are to be
# employed.  Also realize that an SSSC's applied will be relative to ellipticity, elevation and bulk
# static station correction terms.
#
# Sta                 Phase     Velocity     Sed.     Bulk
# Name                Type      Model        Vel.     Station    Comments
#                                            (km/s)   Corr (s)
APA0                  Pn        ak135        5.8       0.0
APA0                  Pg        ak135        5.8       0.0
APA0                  Sn        ak135        3.35      0.0
APA0                  Lg        ak135        3.35      0.0
APZ9                  Pn        ak135        5.8       0.0
APZ9                  Pg        ak135        5.8       0.0
APZ9                  Sn        ak135        3.35      0.0
APZ9                  Lg        ak135        3.35      0.0
ARA0                  Pn        ak135        5.8       0.0
ARA0                  Pg        ak135        5.8       0.0
```

ARA0	Sn	ak135	3.35	0.0
ARA0	Lg	ak135	3.35	0.0
ENN	Pn	fenno	5.8	0.0
ENN	Pg	fenno	5.8	0.0
ENN	Sn	fenno	3.35	0.0
ENN	Lg	fenno	3.35	0.0
FIA0	Pn	ak135	5.8	-0.3
FIA0	Pg	ak135	5.8	-0.5
FIA0	Sn	ak135	3.35	-0.7
FIA0	Lg	ak135	3.35	-0.6
FIA1	Pn	fenno	5.8	0.0
FIA1	Pg	fenno	5.8	0.0
FIA1	Sn	fenno	3.35	0.0
FIA1	Lg	fenno	3.35	0.0
GEC2	Pn	ak135	5.8	0.0
GEC2	Pg	ak135	5.8	0.0
GEC2	Sn	ak135	3.35	0.0
GEC2	Lg	ak135	3.35	0.0
GRA1	Pn	ak135	5.8	0.0
GRA1	Pg	ak135	5.8	0.0
GRA1	Sn	ak135	3.35	0.0
GRA1	Lg	ak135	3.35	0.0
HFS	Pn	fenno	5.8	0.0
HFS	Pg	fenno	5.8	0.0
HFS	Sn	fenno	3.35	0.0
HFS	Lg	fenno	3.35	0.0
KAF	Pn	fenno	5.8	0.0
KAF	Pg	fenno	5.8	0.0
KAF	Sn	fenno	3.35	0.0
KAF	Lg	fenno	3.35	0.0
KSP	Pn	ak135	5.8	-0.4
KSP	Pg	ak135	5.8	-0.6
KSP	Sn	ak135	3.35	-1.0
KSP	Lg	ak135	3.35	-1.0
LOR	Pn	fenno	5.8	0.0
LOR	Pg	fenno	5.8	0.0
LOR	Sn	fenno	3.35	0.0
LOR	Lg	fenno	3.35	0.0
NRA0	Pn	ak135	5.8	0.0
NRA0	Pg	ak135	5.8	0.0
NRA0	Sn	ak135	3.35	0.0
NRA0	Lg	ak135	3.35	0.0
OSS	Pn	fenno	5.8	0.0
OSS	Pg	fenno	5.8	0.0
OSS	Sn	fenno	3.35	0.0
OSS	Lg	fenno	3.35	0.0
SQTA	Pn	fenno	5.8	0.0
SQTA	Pg	fenno	5.8	0.0
SQTA	Sn	fenno	3.35	0.0
SQTA	Lg	fenno	3.35	0.0
VAF	Pn	fenno	5.8	0.0
VAF	Pg	fenno	5.8	0.0
VAF	Sn	fenno	3.35	0.0
VAF	Lg	fenno	3.35	0.0
VRAC	Pn	ak135	5.8	0.0
VRAC	Pg	ak135	5.8	0.0
VRAC	Sn	ak135	3.35	0.0
VRAC	Lg	ak135	3.35	0.0
YKA	Pn	fenno	5.8	0.0
YKA	Pg	fenno	5.8	0.0
YKA	Sn	fenno	3.35	0.0
YKA	Lg	fenno	3.35	0.0

Appendix H Sample Travel-Time Table

This appendix contains a sample travel-time table (velocity model AK135; phase Pn) which contains both two-dimensional distance vs. depth travel-times (seconds) as well as travel-time modelling error table. Note that in this case the modelling errors are distance dependent (as described in Table 13; Appendix C).

```
# ak135 travel-time table for phase: Pn
 12   Number of depth samples at the following depths (km):
    0.00    5.00   15.00   35.00   50.00   75.00  100.00  150.00  200.00  250.00
  300.00  350.00
 41   Number of distance samples at the following distances (deg):
    1.00    1.50    2.00    2.50    3.00    3.50    4.00    4.50    5.00    5.50
    6.00    6.50    7.00    7.50    8.00    8.50    9.00    9.50   10.00   10.50
   11.00   11.50   12.00   12.50   13.00   13.50   14.00   14.50   15.00   15.50
   16.00   16.50   17.00   17.50   18.00   18.50   19.00   19.50   20.00   20.50
   21.00
# Travel time at depth =   0.00 km.
     21.2729
     28.1499
     35.0267
     41.9031
     48.7790
     55.6541
     62.5284
     69.4017
     76.2738
     83.1445
     90.0138
     96.8814
    103.7473
    110.6111
    117.4729
    124.3324
    131.1894
    138.0439
    144.8956
    151.7445
    158.5903
    165.4329
    172.2721
    179.1078
    185.9398
    192.7679
    199.5920
    206.4121
    213.2280
    219.8614
    226.3710
    232.7953
    239.1266
    245.3825
    251.5747
    257.7083
    263.7837
    269.8008
    275.7595
    281.6591
    287.4987
# Travel time at depth =   5.00 km.
     20.6726
     27.5496
        .
        .
      etc.
# Distance-dependent modelling error(s)
   4    1
    0.00   16.00   18.00   21.00
#
   0.100
   3.000
   0.700
   0.700
```

Appendix I Sample Long-Period T-T Table

This appendix contains a sample long-period grid and group velocity file. The first sample file is an LR grid file (as described in Table 16; Appendix D). The second sample file (next page) is the associated LR group velocity file (as described in Table 17; Appendix D).

```
# #Lats  #Lons  Spacing (deg.) for LR phase velocity grid indicies
    18     36     10.000

# Co-Latitude Range:   0.000 to  10.000 deg.
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1

# Co-Latitude Range:  10.000 to  20.000 deg.
  1  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  1
  1  1  4  4  4  4  4  1  0  0  1

# Co-Latitude Range:  20.000 to  30.000 deg.
  4  4  4  4  4  4  4  4  4  4  4  4  5  4  4  5  5  4  5  5  5  5  5  5  4
  4  4  4  4  4  4  4  1  0  0  1

# Co-Latitude Range:  30.000 to  40.000 deg.
  4  4  4  4  4  4  4  4  4  4  5  5  5  5  5  6  6  6  6  6  6  1  1  5  5
  4  4  4  4  4  2  2  1  1  2  2

# Co-Latitude Range:  40.000 to  50.000 deg.
  5  5  5  5  5  5  5  5  5  5  5  5  5  5  6  3  3  2  2  2  2  1  1  1  5
  5  4  4  4  4  2  2  1  1  2  5

# Co-Latitude Range:  50.000 to  60.000 deg.
  5  5  6  5  5  5  5  5  5  5  5  5  5  6  6  3  3  3  3  2  2  2  1  1  5
  5  4  4  3  3  3  2  1  1  2  5

# Co-Latitude Range:  60.000 to  70.000 deg.
  4  4  4  5  5  4  4  4  5  5  5  5  6  1  6  3  3  3  2  2  2  2  1  1  1
  5  5  5  3  3  2  1  0  1  2  4

# Co-Latitude Range:  70.000 to  80.000 deg.
  4  4  4  5  5  0  4  4  2  5  5  1  6  2  6  3  3  3  3  3  2  2  2  1  1
  0  6  6  6  6  2  1  1  2  4  4

# Co-Latitude Range:  80.000 to  90.000 deg.
  4  4  4  5  5  1  2  2  2  6  5  5  6  2  1  3  3  3  3  3  2  2  1  1  1
  0  0  0  5  5  4  2  1  1  2  4

# Co-Latitude Range:  90.000 to 100.000 deg.
  2  4  4  5  2  1  1  1  2  2  6  6  6  5  6  6  3  3  3  3  2  2  1  1  0
  0  0  0  5  4  4  4  2  1  1  1

# Co-Latitude Range: 100.000 to 110.000 deg.
  2  2  4  5  2  1  1  1  2  2  3  3  4  4  5  2  6  2  6  2  2  2  1  1  0
  0  0  1  5  4  4  4  2  1  0  1

# Co-Latitude Range: 110.000 to 120.000 deg.
  2  2  4  2  5  2  1  1  2  2  3  4  4  4  5  2  5  1  6  2  2  2  1  0  0
  0  0  1  2  4  4  2  2  1  0  1

# Co-Latitude Range: 120.000 to 130.000 deg.
  2  2  2  2  1  0  0  0  1  1  2  2  2  4  5  2  5  6  2  2  2  2  1  1  0
  0  0  0  1  4  3  2  2  1  0  1

                                         .
                                         .
                                         .
                                        etc.
```

```
# LR velocity table discretized as index/period (compliments of Mark Woods)
  7     # number of index samples
 28     # number of period samples
   17.00   18.00   20.00   22.00   24.00   26.00   29.00   32.00   35.00   38.00
   42.00   46.00   51.00   56.00   63.00   70.00   78.00   87.00   90.00  111.00
  127.00  145.00  167.00  193.00  223.00  259.00  300.00  400.00
# Velocity for index sample  0
     3.588
     3.620
     3.704
     3.762
     3.798
     3.821
     3.843
     3.847
     3.845
     3.836
     3.819
     3.799
     3.772
     3.746
     3.714
     3.690
     3.670
     3.657
     3.649
     3.642
     3.632
     3.617
     3.590
     3.554
     3.524
     3.541
     3.669
     4.350
# Velocity for index sample  1
     3.599
     3.694
     3.820
     3.899
     3.950
     3.979
     4.002
     4.008
     4.005
     3.995
     3.976
     3.953
     3.927
     3.899
     3.863
     3.831
     3.799
     3.770
     3.743
      .
      .
     etc.
```

Appendix J Glossary/Acronym List

This appendix defines specific terminology used in this document.

TERM	DEFINITION
ADSN	AFTAC Distributed Sub-System Network
AFTAC	Air Force Technical Applications Center
ARS	Analyst Review Station
CCB	Configuration Control Board
CMR	Center for Monitoring Research
CTBT	Comprehensive Test Ban Treaty
DBMS	Database Management System
CTBT	Comprehensive Test Ban Treaty
EvLoc	Event Location (and Magnitude) Program
GSE	Group of Scientific Experts
IASPEI	International Association of Seismology and Physics of the Earth's Interior
IDC	International Data Center
KA	Knowledge Acquisition
LocSAT	Event location using Slowness, Azimuth and Travel-Time data
par files	Input parameter files of the form defined by [getpar(3)]
PIDC	Prototype International Data Center
RMS	Root-mean square
SRST	Source-region station term as defined by Veith (1975)
SSSC	Source-specific station correction
VMSF	Velocity model specification file

Appendix K References

This appendix identifies documents that may contain useful information about technical references, program maintenance and database details.

Anderson, J., W. Farrell, K. Garcia, J. Given, and H. Swanger, "Center for Seismic Studies Database: Version 3.0 Structure" *Science Applications International Corporation Technical Report*, June 1990.

Bratt, S. and T. Bache., "Locating events with a sparse network of regional arrays," *Bull. Seism. Soc. Am.*, 78, 780-798, 1988.

Dziewonski, A. and F. Gilbert., "The effect of small, aspherical perturbations on travel times and a re-examination of the correction for ellipticity," *Geophys. J. R. astr. Soc.*, 44, 7-17, 1976.

Grant, L. and J. Coyne, "Ground-truth data for seismic discrimination research," *Proceedings from the 14th Annual PL/DARPA Seismic Research Symposium*, Tucson, AZ, PL-TR-92-2210, 139-145, 1992

Jenkins, R., W. Nagy, T. Sereno, H. Swanger and D. Wahl, "Regional attenuation and travel-time characteristics at GERESS," *EOS Trans., Am. Geophys. Un.*, 73, 374, 1992.

Kennett, B., "IASPEI 1991 Seismological Tables," *Research School of Earth Sciences, Australian National University*, 167 pp., 1991.

Kennett, B., and E. Engdahl, "Traveltimes for global earthquake location and phase identification," *Geophys. J. Int.*, 105, 429-465, 1991.

Kennett, B., E. Engdahl, and R. Buland, "Constraints on seismic velocities in the Earth from traveltimes," *Geophys. J. Int.*, in press, 1996.

Kernighan, B. and D. Ritchie., "The C Programming Language (2nd edition)," *Prentice Hall Software Series.*, Prentice Hall, Englewood Cliffs, N.J., 272 pp., 1988.

Sambridge, M., J. Braun and H. McQueen, "Geophysical parameterization and interpolation of irregular data using natural neighbors," *Geophys. J. Int.*, in press, 1995.

Sambridge, M. and B. Kennett, "A novel method of hypocentre location," *Geophys. J. Int.*, 87, 1986.

SAIC, "Intelligent Array System: Software Policies and Procedures, Rev. 2.3.9" *Science Applications International Corporation Technical Report SAIC-88/1596*, March 1988.

Swanger, H., J. Anderson, T. J. Sereno, Jr., J. Given, and D. Williams, "Extensions to the Center Version 3 Database" *Special Technical Report SAIC-93/1123*, October 1993.

Veith, K., "Refined hypocenters and accurate reliability estimates," *Bull. Seism. Soc. Am.*, 65, 1199-1222, 1975.

UNIX manual pages:

ARS(1)	UNIX manual page for ARS program.
EvLoc(1)	UNIX manual page for EvLoc program.
LocSAT(1)	UNIX manual page for LocSAT program.
libloc(3)	UNIX manual page for libloc library routines.